

ระบบจัดการเมลถึงลิสต์ย่อย

SUBMAILING LIST MANAGEMENT SYSTEM



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ  
บัณฑิตวิทยาลัย  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2543

เลขหมู่.....  
เลขทะเบียน..... 35934  
วัน, เดือน, ปี..... 3 ก.ค. 2543

ISBN 974-622-818-8

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SUBMAILING LIST MANAGEMENT SYSTEM**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
MASTER OF SCIENCE PROGRAM IN COMPUTER SCIENCE AND  
INFORMATION TECHNOLOGY  
SCHOOL OF GRADUATE STUDIES.  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2000**

**ISBN 974-622-818-8**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**COPYRIGHT 2000**

**SCHOOL OF GRADUATE STUDIES**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	ระบบจัดการเมลลิงลิสต์ย่อย
นักศึกษา	นายสรวิศ บุญมี
รหัสประจำตัว	36064026
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
พ.ศ.	2543
อาจารย์ผู้ควบคุมวิทยานิพนธ์	อาจารย์สุรสิทธิ์ วรรณไกรโรจน์
อาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม	ดร.จันทร์บูรณ์ สถิตวิริยวงศ์

### บทคัดย่อ

เมลลิงลิสต์เป็นการประยุกต์ใช้ไปรษณีย์อิเล็กทรอนิกส์โดยใช้ประโยชน์จากความสามารถในการส่งอีเมลเพียงฉบับเดียวถึงผู้รับหลายๆ คน แต่เมลลิงลิสต์อาจก่อให้เกิดปัญหาในด้านความซ้ำซ้อนในการส่ง และความซ้ำซ้อนของเนื้อหาที่ใช้เก็บจดหมาย ที่ส่งถึงสมาชิกที่อยู่ในเครือข่ายเดียวกันหรือใกล้เคียงกัน นอกจากนี้ยังอาจก่อให้เกิดการตกค้างสะสมของจดหมายเก่าๆ จากเมลลิงลิสต์ที่สมาชิกไม่มีเวลาจัดการ วิทยานิพนธ์ฉบับนี้ เสนอวิธีการแก้ปัญหาของเมลลิงลิสต์เหล่านี้ โดยการแบ่งกลุ่มสมาชิกออกเป็นเมลลิงลิสต์ย่อยๆ ตามความใกล้เคียงของเครือข่าย เพื่อลดจำนวนจดหมายจากเมลลิงลิสต์ที่ส่งระหว่างเครือข่ายลงเหลือเพียงสำเนาเดียวต่อหนึ่งเมลลิงลิสต์ย่อย และใช้โปรโทคอล MIME แก้ปัญหาความซ้ำซ้อนของเนื้อหาที่ใช้เก็บจดหมายและการตกค้างสะสมของจดหมายเก่า โดยเก็บจดหมายต้นฉบับไว้ที่ส่วนกลางเพียงสำเนาเดียว แล้วจึงส่งจดหมายที่เป็นตัวซ้ำบอกวิธีที่จะเข้าถึงจดหมายต้นฉบับนั้นให้แก่สมาชิกของเมลลิงลิสต์ย่อย และคอยลบจดหมายต้นฉบับเก่าที่ส่วนกลางทิ้งเป็นระยะๆ

<b>Thesis Title</b>	Submailing List Management System
<b>Student</b>	Mr.Sorawit Boonmee
<b>Student ID.</b>	36064026
<b>Degree</b>	Master of Science
<b>Programme</b>	Computer Science and Information Technology
<b>Year</b>	2000
<b>Thesis Advisor</b>	Mr.Surasit Vannakrairojn
<b>Thesis Co-advisor</b>	Dr.Jantaboon Satitviriyavong

## ABSTRACT

Mailing lists represent another way to apply electronic mail by using ability to send an only one message to a list of many recipients. But the problems of mailing lists are the redundancy of mails that send across network to subscribers in the same domain or the neighbors and the redundancy of storage for those mails. Another problem is the accumulation of many old mails from the mailing lists in subscriber that have no time to manage them. This thesis proposes a solution to these problems by splitting up the mailing list into chunks of a submailing list based on network proximity of subscribers. This solution will decrease the number of mails from a mailing list to only one copy per submailing list. This thesis also use MIME protocol to solve the storage redundancy and the accumulation of old mails problem by storing the original mail at a center host and sending a mail to subscribers for telling how to access this original mail. And the old mails at the center host will be deleted frequently.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้เป็นอย่างดี ด้วยคำแนะนำและการให้คำปรึกษาของ อาจารย์สุรสิทธิ์ วรรณไกรโรจน์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ และดร.จันทร์บูรณ์ สถิตวิริยวงศ์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ร่วม ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์ของท่านทั้งสอง จึงขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณอาจารย์เอื้อน ปิ่นเงิน เป็นอย่างสูง ที่ได้ช่วยเหลือให้คำแนะนำในเรื่อง การตีพิมพ์บทความวิจัยอันเกิดจากวิทยานิพนธ์นี้

ขอขอบพระคุณอาจารย์อักรินทร์ คุณกิตติ ซึ่งเป็นผู้ริเริ่มเสนอแนวคิดอันเป็นที่มาของ วิทยานิพนธ์นี้

ขอขอบพระคุณเจ้าหน้าที่คณะเทคโนโลยีสารสนเทศทุกท่าน ที่คอยให้คำปรึกษา และ อำนวยความสะดวกแก่ผู้วิจัย

ขอขอบคุณเพื่อนๆ นักศึกษาทุกท่าน ที่คอยช่วยเหลือให้คำแนะนำ อำนวยความสะดวก และเป็นกำลังใจให้กับผู้วิจัยตลอดมา

สุดท้ายขอขอบคุณบัณฑิตวิทยาลัย ที่ได้ให้ทุนสนับสนุนการทำวิทยานิพนธ์นี้

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอบแต่ผู้มีพระคุณทุกท่าน

สรวิศ บุญมี

# สารบัญ

หน้า

บทคัดย่อภาษาไทย .....	I
บทคัดย่อภาษาอังกฤษ .....	II
กิตติกรรมประกาศ.....	III
สารบัญ .....	IV
สารบัญตาราง .....	VII
สารบัญรูป .....	VIII
บทที่ 1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมุติฐานของการศึกษา.....	2
1.4 ขอบเขตของการศึกษา.....	2
1.5 ขั้นตอนของการศึกษา.....	4
1.6 ข้อตกลงเบื้องต้น .....	4
1.7 คำจำกัดความที่ใช้ในการศึกษา.....	4
บทที่ 2 ไปรษณีย์อิเล็กทรอนิกส์และเมลลิงลิสต์.....	5
2.1 กลไกของระบบไปรษณีย์อิเล็กทรอนิกส์.....	5
2.2 MUA กับ MTA .....	6
2.3 Simple Mail Transfer Protocol (SMTP).....	7
2.4 รูปแบบของไปรษณีย์อิเล็กทรอนิกส์ .....	9
2.5 เมลลิงลิสต์.....	10
2.6 การจัดการการเมลลิงลิสต์.....	10
2.7 ซอฟต์แวร์จัดการเมลลิงลิสต์ (Mailing List Manager).....	11
2.8 สมรรถภาพในการกระจายจดหมายของซอฟต์แวร์จัดการเมลลิงลิสต์.....	13
2.9 การแก้ปัญหาความซ้ำซ้อนของจดหมายจากเมลลิงลิสต์โดยใช้ลิสต์ย่อย.....	14
2.9.1 เมลลิงลิสต์ย่อย .....	14
2.9.2 การทำงานของซอฟต์แวร์จัดการเมลลิงลิสต์ที่เป็นอุปสรรคต่อเมลลิงลิสต์ย่อย ....	16
2.9.3 การสร้างเมลลิงลิสต์ย่อย .....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า

บทที่ 3 การใช้โปรโทคอล MIME ในการแก้ปัญหาของเมลลิงลิสต์ .....	19
3.1 บทนำ.....	19
3.2 ความเป็นมาของโปรโทคอล MIME .....	19
3.3 MIME-Version Header .....	20
3.4 Content-ID Header.....	21
3.5 Content-Type Header.....	21
3.6 Media Types.....	22
3.6.1 RFC822 Subtype.....	23
3.6.2 External-Body Subtype.....	26
3.7 สรุป .....	28
บทที่ 4 การออกแบบและการทำงานของระบบจัดการเมลลิงลิสต์ย่อย.....	29
4.1 Requirement Specification.....	29
4.2 Systems Analysis .....	29
4.2.1 Data Flow Analysis.....	29
4.2.2 Data Dictionary.....	32
4.3 การออกแบบระบบ.....	34
4.3.1 โครงสร้างของระบบ.....	34
4.3.2 สภาพแวดล้อมของระบบ .....	35
4.3.3 รูปแบบของเพิ่มข้อมูล.....	36
4.4 การทำงานของแต่ละมอดูล.....	38
4.4.1 มอดูล 1.0: Forward Mail between Lists .....	38
4.4.2 มอดูล 2.0: Maintain System .....	45
บทที่ 5 ผลการทดลอง .....	48
5.1 การออกแบบการทดลอง.....	48
5.2 สภาพแวดล้อมของการทดลอง.....	48
5.3 การทดลองปริมาณข้อมูลที่ส่งระหว่างเครือข่าย .....	48
5.3.1 ผลการทดลอง.....	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

หน้า

5.3.2	วิเคราะห์ผลการทดลอง .....	52
5.4	การทดลองเนื้อที่ที่ใช้ในการเก็บจดหมาย.....	52
5.4.1	ผลการทดลอง.....	52
5.4.2	วิเคราะห์ผลการทดลอง .....	56
บทที่ 6	สรุปผลการวิจัยและข้อเสนอแนะ.....	58
6.1	สรุปผลการวิจัย.....	58
6.2	ประโยชน์ของระบบจัดการเมลลิงลิสต์ย่อย.....	58
6.3	ข้อเสนอแนะ.....	59
เอกสารอ้างอิง .....		61
ภาคผนวก ก. วิธีใช้ระบบจัดการเมลลิงลิสต์ย่อย .....		63
ความต้องการของระบบ (System Requirement).....		63
การติดตั้ง.....		63
การใช้งาน .....		64
ภาคผนวก ข. สรุปคำสั่งและคำตอบรับของโปรโตคอล SMTP .....		65
สรุปคำสั่งของโปรโตคอล SMTP.....		65
รหัสและคำตอบรับ .....		66
ภาคผนวก ค. บทความที่ได้รับการตีพิมพ์.....		67
ประวัติผู้เขียน .....		79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
4.1 แสดงความหมายของพารามิเตอร์แต่ละตัวของแฟ้ม Gateway.ini.....	36
5.1 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ Communication Network.....	49
5.2 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ List Managers.....	50
5.3 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ Majordomo Users.....	51
5.4 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ Communication Network.....	53
5.5 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ List Managers.....	54
5.6 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ Majordomo Users.....	55



# สารบัญรูป

รูปที่	หน้า
1.1 โครงสร้างเมลลิงลิสต์ (A) แบบธรรมดาทั่วไป (B) หลังจากเพิ่มลิสต์ย่อยเข้าไป.....	3
2.1 องค์ประกอบของระบบไปรษณีย์อิเล็กทรอนิกส์.....	6
2.2 โมเดลของ SMTP.....	7
2.3 ไปรษณีย์อิเล็กทรอนิกส์ประกอบด้วย Header และ Body .....	9
2.4 แสดงโครงสร้างของเมลลิงลิสต์ (a) แบบปกติ (b) แบบมีลิสต์ย่อย .....	15
2.5 ทิศทางในการถ่ายทอดจดหมาย.....	17
3.1 รูปแบบของจดหมายแบบ Message/RFC822 .....	23
3.2 ตัวอย่างจดหมายต้นฉบับจากเมลลิงลิสต์ย่อย ก่อนที่จะถูกแปลงด้วย Sublist gateway .....	24
3.3 ตัวอย่างจดหมายที่ถูกแปลงด้วย Sublist gateway โดยใช้รูปแบบ Message/RFC822 .....	25
3.4 รูปแบบของจดหมายแบบ Message/External-body.....	26
3.5 ตัวอย่างจดหมายที่ถูกแปลงด้วย Sublist gateway เป็นรูปแบบ Message/External-Body .....	27
4.1 Context diagram ของระบบจัดการเมลลิงลิสต์ย่อย .....	30
4.2 Level-0 Data Flow Diagram ของระบบจัดการเมลลิงลิสต์ย่อย.....	31
4.3 Level-1 Data Flow Diagram ของขั้นตอนส่งต่อจดหมายระหว่างเมลลิงลิสต์.....	31
4.4 Level-2 Data Flow Diagram ของขั้นตอนบำรุงรักษาระบบ.....	32
4.5 Structure Chart ของระบบจัดการเมลลิงลิสต์ย่อย .....	34
4.6 ผังงาน โปรแกรมของมอดูล 1.0: Forward Mail between Lists.....	38
4.7 ผังงาน โปรแกรมของมอดูล 1.1: Get Mail.....	39
4.8 ผังงาน โปรแกรมของมอดูล 1.2: Check for Duplicate Mail by its Message-Id.....	40
4.9 ผังงาน โปรแกรมของมอดูล 1.3: Check Recipient .....	41
4.10 ผังงาน โปรแกรมของมอดูล 1.4: Process Mail to Main List .....	43
4.11 ผังงาน โปรแกรมของมอดูล 1.5: Process Mail to Sublist.....	44
4.12 ผังงาน โปรแกรมของมอดูล 2.0: Maintain System.....	46
4.13 ผังงาน โปรแกรมของมอดูล 2.1: Delete Expired Message-Id .....	46
4.14 ผังงาน โปรแกรมของมอดูล 2.2: Delete Expired Mail .....	47
5.1 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ Communication Network.....	49
5.2 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ List Managers.....	50
5.3 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ Majordomo Users.....	51
5.4 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ Communication Network .....	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
5.5 เปรียบเทียบเนื้อหาที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ List Managers.....	54
5.6 เปรียบเทียบเนื้อหาที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ Majordomo Users.....	55
5.7 เปรียบเทียบเนื้อหาที่คัดได้จากการทดลองกับเมลลิงลิสต์ทั้งสามและการวิเคราะห์.....	56



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ไปรษณีย์อิเล็กทรอนิกส์ หรือเรียกย่อๆ ว่า อีเมล (Electronic Mail; E-Mail) เป็นบริการหรือการประยุกต์ใช้เครือข่ายอินเทอร์เน็ตที่เก่าแก่ดั้งเดิมมาตั้งแต่เริ่มก่อตั้งอินเทอร์เน็ต และยังคงมีความสำคัญเป็นที่รู้จัก และมีผู้ใช้มากที่สุดอย่างหนึ่งมาจนถึงทุกวันนี้ ผู้คนนับล้านๆ คนทั่วโลกที่ใช้อินเทอร์เน็ตจะต้องมีอีเมลแอดเดรสเป็นของตัวเอง

เมลลิงลิสต์ (Mailing List) หรือ บัญชีจำหน่าย เป็นการประยุกต์ใช้ไปรษณีย์อิเล็กทรอนิกส์รูปแบบหนึ่ง โดยใช้ประโยชน์จากความสามารถในการส่งอีเมลเพียงฉบับเดียวถึงผู้รับหลายๆ คนได้โดยง่ายเพียงจำหน่ายถึงผู้รับเหล่านั้นด้วยอีเมลแอดเดรสเพียงชื่อเดียว เมลเซิร์ฟเวอร์ก็จะทำสำเนาอีเมลฉบับนั้นส่งให้ผู้รับทุกคนโดยอัตโนมัติ ด้วยเหตุนี้จึงมีการนำเมลลิงลิสต์มาประยุกต์ใช้ในการประชุม ปรัชญาหรือ แลกเปลี่ยนความคิดเห็น หรือเพื่อแจ้งข่าวสารต่างๆ ให้แก่กลุ่มสมาชิก

เนื่องจากระบบเมลลิงลิสต์จะส่งจดหมายถึงผู้ใช้ทุกคน แม้ว่าผู้ใช้เหล่านั้นจะอยู่บนเครือข่ายย่อย โดเมน หรือ แม้ข่ายเดียวหรือใกล้เคียงกัน ทำให้เกิดความซ้ำซ้อนของข้อมูล โดยเฉพาะอย่างยิ่งถ้ามีสมาชิกของลิสต์เดียวกันอยู่ภายในเครือข่ายใกล้เคียงกันเป็นจำนวนมาก ทำให้การจราจรระหว่างเครือข่ายสูงขึ้นโดยไม่จำเป็น นอกจากนี้ยังต้องสิ้นเปลืองเนื้อที่ในการเก็บจดหมายเหล่านี้ในสมาชิกแต่ละคนซ้ำๆ กัน และส่วนใหญ่สมาชิกเหล่านี้จะไม่สามารถอ่านจดหมายจากเมลลิงลิสต์ที่มีสมาชิกจำนวนมากได้ในวันเดียว ซึ่งในวันต่อไปก็จะมีจดหมายจำนวนมากพอกันที่ไม่สามารถอ่านหมดได้เพิ่มขึ้นเรื่อยๆ ทำให้เกิดการสะสมของจดหมายที่ซ้ำซ้อนกันเป็นจำนวนมากใน Mailbox ของแต่ละคน ซึ่งเป็นการใช้ทรัพยากรของแม่ข่ายที่ใช้เก็บจดหมายเหล่านี้อย่างไม่คุ้มค่า [1]

### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

จากปัญหาที่กล่าวมาข้างต้น ความมุ่งหมายและวัตถุประสงค์ของการศึกษาคือ

1. ทำอย่างไรจึงจะกำจัดความซ้ำซ้อนของจดหมายที่ส่งมาให้ผู้รับที่อยู่ภายในเครือข่ายใกล้เคียงกัน เพื่อลดการจราจร และเพิ่มประสิทธิภาพในการส่งข้อมูลระหว่างเครือข่าย
2. หาวิธีที่จะกำจัดความซ้ำซ้อนในการเก็บจดหมายจากเมลลิงลิสต์ของสมาชิกแต่ละคน โดยไม่ต้องแก้ไขโปรแกรมที่สมาชิกใช้อ่านจดหมาย (Mail User Agent หรือ MUA)
3. แก้ปัญหาการตกค้างสะสมของจดหมายใน Mailbox ของสมาชิกแต่ละคน เพื่อให้เป็นการใช้ทรัพยากรของแม่ข่ายที่ใช้เก็บจดหมายเหล่านี้ได้อย่างคุ้มค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 สมมุติฐานของการศึกษา

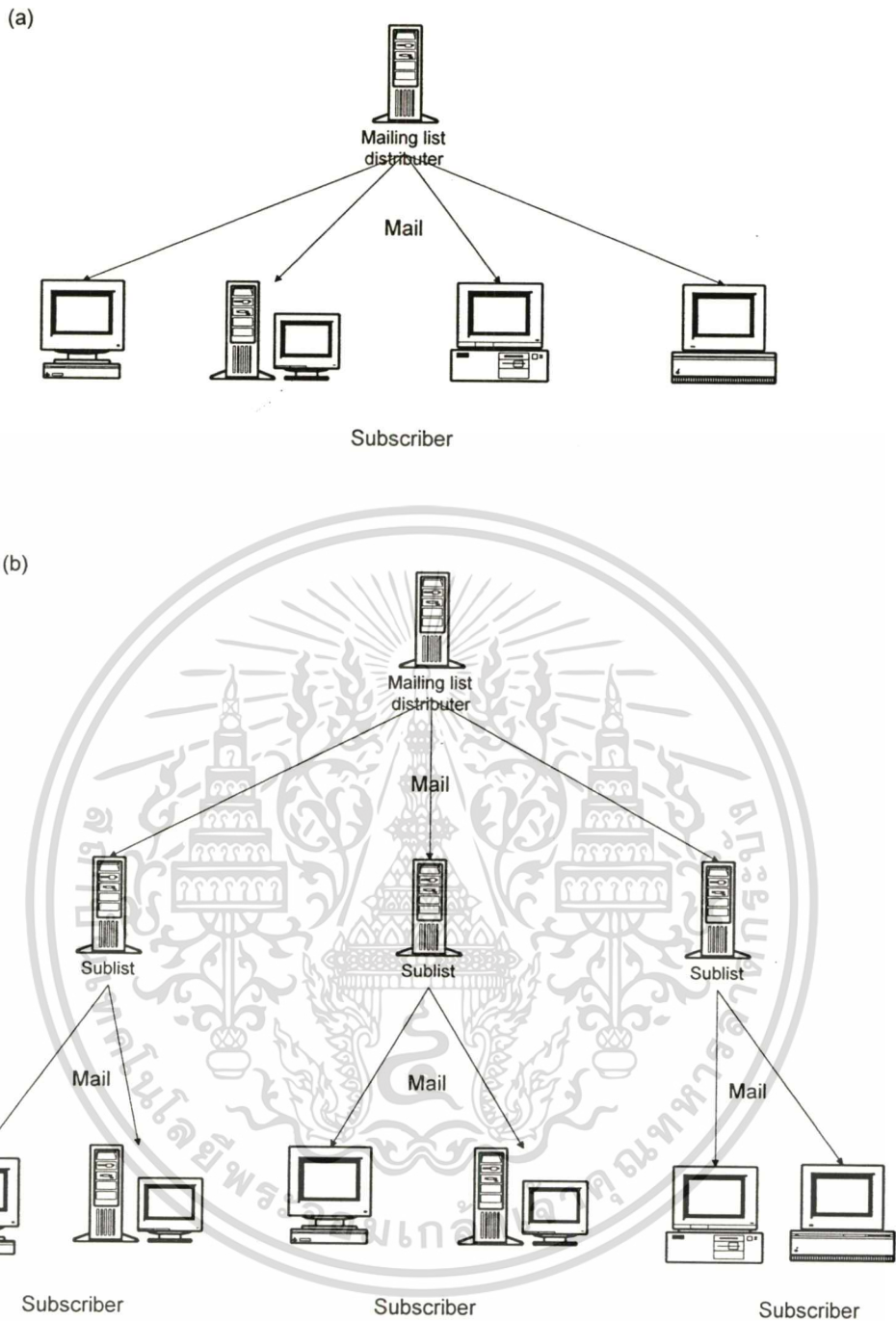
จากความมุ่งหมายและวัตถุประสงค์ของการศึกษาที่กล่าวมาข้างต้น สมมุติฐานของการศึกษาเพื่อให้บรรลุถึงความมุ่งหมายและวัตถุประสงค์ในการแก้ปัญหาคือ

1. แบ่งกลุ่มสมาชิกของเมลลิงลิสต์ที่อยู่ในเครือข่ายเดียวกันหรือใกล้เคียงกันออกเป็นเมลลิงลิสต์ย่อยๆ (Submailing List หรือ Sublist หรือ Exploder [2]) ซึ่งคือแอ็ดเดรสของสมาชิกที่แทนที่จะเป็นอีเมลแอ็ดเดรสธรรมดา กลับกลายเป็นแอ็ดเดรสของเมลลิงลิสต์อีกลิสต์หนึ่ง วิธีนี้จะกำจัดความซ้ำซ้อนของจดหมายที่จะถูกส่งให้สมาชิกลงไปได้ เพราะแทนที่จะต้องส่งสำเนาจดหมายถึงสมาชิกทุกคนในเครือข่ายที่ใกล้เคียงกัน ก็จะเปลี่ยนมาเป็นการส่งเพียงสำเนาเดียวให้กับลิสต์ย่อยของเครือข่ายใกล้เคียง จากนั้นลิสต์ย่อยก็จะกระจายต่อให้สมาชิกของเครือข่ายใกล้เคียงต่อไป ซึ่งจะช่วยลดการจราจรของข้อมูลระหว่างเครือข่ายได้ ดังรูปที่ 1.1 ซึ่งแสดงโครงสร้างเปรียบเทียบกันระหว่างเมลลิงลิสต์แบบธรรมดาที่ไม่มีลิสต์ย่อย (A) กับแบบที่มีลิสต์ย่อย (B)
2. เก็บจดหมายจากเมลลิงลิสต์หลักไว้ที่ส่วนกลาง จากนั้นจึงส่งจดหมายที่มีลักษณะเป็นเพียงตัวชี้ (Pointer) ให้แก่สมาชิกเพื่อจะบอก Mail User Agent ของสมาชิกถึงวิธีในการดึงเอาตัวจดหมายที่แท้จริงจากส่วนกลางมาแสดง ซึ่งจะทำให้ไม่เกิดการเก็บจดหมายที่ซ้ำซ้อนกันในสมาชิกแต่ละคน โดยอาศัย MIME (Multipurpose Internet Mail Extensions) ซึ่งเป็นโพรโทคอลที่ช่วยขยายรูปแบบของไปรษณีย์อิเล็กทรอนิกส์ซึ่งแต่เดิมกำหนดไว้เพียงให้ใช้ข้อความที่เป็นชุดอักขระ US-ASCII ให้ยอมรับรูปแบบอื่นมากขึ้น เช่น สามารถใช้ชุดอักขระอื่นนอกเหนือจาก US-ASCII หรือเป็นข้อมูลในรูปแบบอื่นที่ไม่ใช่ข้อความ เช่น รูปภาพ หรือ เพิ่มข้อมูลที่ไม่ใช่ข้อความเป็นต้น [3] โดยใช้ Content-Type แบบ Message/External-Body
3. ลบจดหมายที่ส่วนกลางทิ้งเมื่อครบระยะเวลาที่กำหนด เพื่อคืนเนื้อที่ที่ใช้เก็บจดหมายเหล่านี้ และไม่ทำให้เกิดการตกค้างสะสมของจดหมายเก่าๆ จากเมลลิงลิสต์ของสมาชิกแต่ละคน

### 1.4 ขอบเขตของการศึกษา

จากสมมุติฐานของการศึกษาข้างต้น สามารถกำหนดขอบเขตของการศึกษาได้ดังนี้

1. ศึกษาเกี่ยวกับบริการไปรษณีย์อิเล็กทรอนิกส์ที่ใช้สำหรับอินเทอร์เน็ตอย่างละเอียด ทั้งองค์ประกอบ รูปแบบ และโพรโทคอลที่ใช้ในการรับส่ง (SMTP) เพื่อให้เข้าใจถึงการทำงานของไปรษณีย์อิเล็กทรอนิกส์โดยรวม
2. ศึกษาเกี่ยวกับการทำงานของเมลลิงลิสต์ทั้งในแง่มุมมองของผู้ใช้ และผู้ดูแลระบบ รวมถึงการทำงานของซอฟต์แวร์ที่ช่วยในการจัดการเมลลิงลิสต์ (Mailing List Manager หรือ MLM) ต่างๆ



รูปที่ 1.1 โครงสร้างเมลลิงลิสต์ (A) แบบธรรมดาทั่วไป (B) หลังจากเพิ่มลิสต์ย่อยเข้าไป

3. ศึกษาการทำงานของโปรโตคอล MIME โดยเฉพาะส่วนที่จะสามารถนำมาใช้เพื่อแก้ปัญหาตามสมมุติฐานข้างต้นอันได้แก่ Content-Type แบบ Message/External-Body และ Message/Rfc822
4. การศึกษาเกี่ยวกับไปรษณีย์อิเล็กทรอนิกส์และเมลลิงลิสต์ จะเน้นเฉพาะระบบที่ทำงานกับระบบปฏิบัติการยูนิกซ์เท่านั้น เนื่องจากเป็นระบบเปิด และเป็นที่ยอมรับใช้อย่างแพร่หลายในการให้บริการอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.5 ขั้นตอนของการศึกษา

ขั้นตอนของการศึกษาโดยสรุปมีรายละเอียดดังนี้

1. ศึกษาการทำงานของส่วนที่เกี่ยวข้องกับเมลลิงลิสต์ อันได้แก่ การทำงานของไพรอซีอีอิเล็กทรอนิกส์ เมลลิงลิสต์และซอฟต์แวร์ที่ใช้ในการจัดการ และโพรโทคอล MIME ในส่วนที่จะนำมาใช้ในการแก้ปัญหา
2. ออกแบบระบบที่จะใช้จัดการเมลลิงลิสต์ย่อยตามสมมุติฐานของการศึกษาที่ตั้งขึ้นมา เลือกสภาพแวดล้อมที่ระบบนั้นจะทำงานอยู่ภายใต้ และเลือกเครื่องมือที่จะใช้ในการพัฒนาระบบ
3. ทำการพัฒนาระบบตามที่ได้ออกแบบมาภายใต้สภาพแวดล้อมและใช้เครื่องมือที่เลือก
4. นำระบบที่ได้มาทดลองเพื่อนำผลมาพิสูจน์สมมุติฐานของการศึกษาที่ตั้งขึ้นมา
5. วิเคราะห์ วิจารณ์ และสรุปผลการทดลองที่ได้

## 1.6 ข้อตกลงเบื้องต้น

ไพรอซีอีอิเล็กทรอนิกส์ที่ศึกษาในวิทยานิพนธ์นี้จำกัดอยู่แต่เพียงไพรอซีอีอิเล็กทรอนิกส์ที่ใช้สำหรับเครือข่ายอินเทอร์เน็ตเท่านั้น

## 1.7 คำจำกัดความที่ใช้ในการศึกษา

คำว่า “จดหมาย” ในวิทยานิพนธ์นี้ถ้าไม่ได้ระบุเจาะจงลงไป ให้มีความหมายแทนไพรอซีอีอิเล็กทรอนิกส์ที่ใช้สำหรับเครือข่ายอินเทอร์เน็ต

## บทที่ 2

### โปรยชน์ย้อเล็กทรอนิกส์และเมลลิ่งลิสต์

โปรยชน์ย้อเล็กทรอนิกส์เป็นบริการที่เก่าแก่ดั้งเดิมมาตั้งแต่เริ่มก่อตั้งอินเทอร์เน็ตที่ยังคงได้รับความนิยม และใช้กันอย่างกว้างขวางมาจนถึงทุกวันนี้ โปรยชน์ย้อเล็กทรอนิกส์ยังเป็นหนึ่งในบริการอันดับแรกๆ ที่ผู้ใช้ส่วนใหญ่ใช้เมื่อเข้าถึงอินเทอร์เน็ต ผู้ใช้บางคนอาจใช้อินเทอร์เน็ตเพียงเพื่อรับหรือส่งอีเมลเป็นหลักเท่านั้น โปรยชน์ย้อเล็กทรอนิกส์ประสบความสำเร็จเป็นอย่างมากดังจะเห็นได้ว่าผู้ใช้อินเทอร์เน็ตจำนวนมากวางใจอาศัยมันเป็นเครื่องมือในการติดต่อทางธุรกิจตามปกติ

โปรยชน์ย้อเล็กทรอนิกส์เป็นที่นิยมเพราะว่ามันเป็นวิธีที่สะดวก รวดเร็ว ในการส่งข่าวสาร มันสามารถถูกพลิกแพลงใช้ได้ตั้งแต่การส่งบันทึกสั้นๆ ไปจนถึงข้อความยาวหลายๆ หน้า นอกจากนี้ผู้ใช้จำนวนมากยังส่งแฟ้มข้อมูลด้วยโปรยชน์ย้อเล็กทรอนิกส์มากกว่าโปรแกรมที่ใช้สำหรับถ่ายโอนแฟ้มข้อมูลโดยเฉพาะ (FTP) [4]

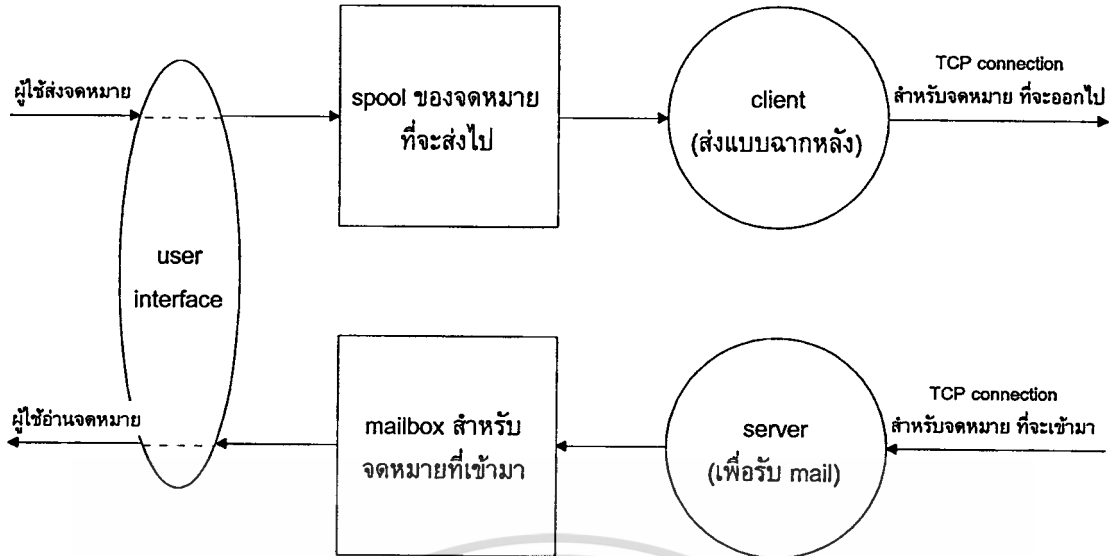
#### 2.1 กลไกของระบบโปรยชน์ย้อเล็กทรอนิกส์

การส่งโปรยชน์ย้อเล็กทรอนิกส์เป็นแนวคิดใหม่เพราะว่ามันแตกต่างจากการใช้เครือข่ายขั้นพื้นฐานอื่นๆ ที่เป็นที่รู้จัก เช่น โพรโทคอลเครือข่ายต่างๆ ไปจะส่งกลุ่มข้อมูล (Packet) โดยตรงสู่ปลายทางโดยใช้การกำหนดช่วงเวลา (Timeout) และการส่งซ้ำสำหรับแต่ละเซกเมนต์ถ้าไม่มีการตอบรับกลับมา ในกรณีของโปรยชน์ย้อเล็กทรอนิกส์ อย่างไรก็ตามระบบต้องเตรียมป้องกันกรณีที่เครื่องปลายทางหรือการเชื่อมโยงเครือข่ายเกิดไม่ทำงาน ผู้ส่งจะไม่ต้องคอยจนเครื่องปลายทางใช้ได้ก่อนที่จะทำงานต่อไป หรือไม่ทำให้ผู้ใช้ต้องยกเลิกการส่งเพียงเพราะการสื่อสารกับเครื่องปลายทางใช้ไม่ได้เพียงชั่วคราว

ในการจัดการกับการส่งที่ล่าช้า ระบบโปรยชน์ย้อเล็กทรอนิกส์จะใช้เทคนิคที่เรียกว่าการ Spool เมื่อผู้ใช้ส่งจดหมาย ระบบจะเก็บมันไว้ใน Spool พร้อมด้วยชื่อผู้ส่ง ผู้รับ เครื่องปลายทาง และ เวลาที่ส่ง จากนั้นระบบจะเริ่มส่งให้แก่เครื่องปลายทางในแบบฉากหลัง ทำให้ผู้ส่งทำงานอื่นต่อไปได้เลย ดังรูปที่ 2.1

กระบวนการที่ส่งจดหมายแบบฉากหลังจะทำหน้าที่เป็นไคลเอนต์ มันจะ Map ชื่อของเครื่องปลายทางเป็นไอพีแอดเดรส และจะพยายามสร้างการเชื่อมต่อแบบ TCP กับตัวเมลเซิร์ฟเวอร์บนเครื่องปลายทาง ถ้าสำเร็จกระบวนการนั้นก็จะส่งข้อความของจดหมายนั้นให้แก่เซิร์ฟเวอร์ปลายทางซึ่งจะเก็บจดหมายที่ได้รับลงใน Spool ของระบบเอง เมื่อไคลเอนต์และเซิร์ฟเวอร์ตกลงยอมรับและเก็บจดหมายนั้นแล้ว ตัวไคลเอนต์จะลบจดหมายของตนทิ้ง ถ้ากระบวนการที่ส่งไม่สามารถสร้างการเชื่อมต่อแบบ TCP หรือถ้าการเชื่อมต่อเกิดใช้การไม่ได้ มันจะบันทึกจำนวนครั้งที่มัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 องค์ประกอบของระบบ ไปรษณีย์อิเล็กทรอนิกส์

พยายามจะส่งและหยุดไป กระบวนการที่ส่งจดหมายแบบฉากหลังจะกวาดตลอด Spool เป็นระยะๆ เพื่อตรวจสอบจดหมายที่ยังไม่ได้ส่ง เมื่อใดก็ตามที่มันพบ หรือ เมื่อใดก็ตามที่ผู้ใช้ส่งจดหมายใหม่ ออกไป มันก็จะพยายามส่งให้อีกครั้ง ถ้ามันพบว่าจดหมายฉบับใดไม่สามารถส่งได้ภายในเวลาที่กำหนด (เช่น สามวัน) มันก็จะตีจดหมายนั้นกลับไปให้ผู้ส่ง [5]

## 2.2 MUA กับ MTA

MUA ย่อมาจาก Mail User Agent คือโปรแกรมที่ผู้ใช้ใช้ อ่าน ตอบ เขียน และส่งอีเมล ตัวอย่างของ MUA คือ /bin/mail mailx pine ซึ่งทำงานบนยูนิกซ์และ Outlook Express ซึ่งทำงานบนวินโดวส์

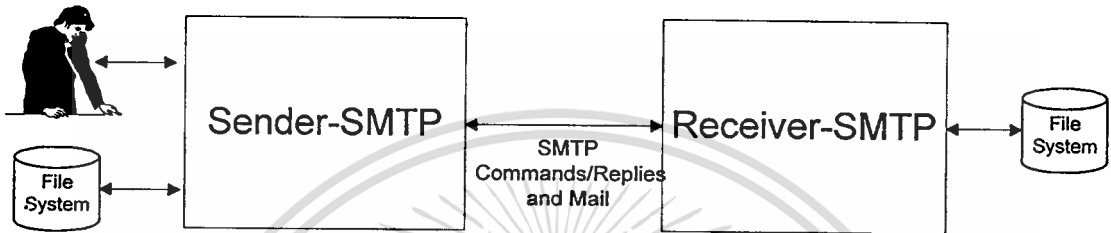
MTA ย่อมาจาก Mail Transport Agent คือโปรแกรมที่ทำหน้าที่ควบคุมการส่งไปรษณีย์อิเล็กทรอนิกส์สำหรับผู้ใช้จำนวนมากๆ และส่งอีเมลต่อ (Forward) ระหว่างเครื่องทั้งหลาย MTA ที่เป็นที่รู้จักกันดีก็คือ sendmail นอกจากนั้นก็ยังยมี MMDF Smail และ Zmailer ซึ่งทั้งหมดทำงานบนยูนิกซ์

MUA จะทำหน้าที่เสมือนปากกาและกระดาษที่ใช้เขียนจดหมาย แต่ละเครื่องอาจมี MUA ได้หลากหลายโปรแกรมตามความต้องการของผู้ใช้ MTA จะทำหน้าที่เสมือนที่ทำการไปรษณีย์ ดังนั้นแต่ละเครื่องหรือเครือข่ายย่อยจึงมักจะมี MTA เพียงตัวเดียว MUA มักจะติดต่อกับ MTA ด้วย โพรโทคอล SMTP (Simple Mail Transfer Protocol) ในกรณีที่อยู่ต่างเครื่องกัน หรือติดต่อกันด้วย IPC (Interprocess Communication) ในกรณีที่อยู่ในเครื่องเดียวกัน

### 2.3 Simple Mail Transfer Protocol (SMTP)

จุดมุ่งหมายของ Simple Mail Transfer Protocol คือการถ่ายโอนไปรษณีย์อิเล็กทรอนิกส์อย่างมีประสิทธิภาพและเชื่อถือได้ [6]

SMTP ได้รับการออกแบบตามโมเดลดังนี้ เมื่อผู้ใช้ส่งจดหมาย โพรเซส SMTP ฝ่ายส่งจะสร้างการเชื่อมต่อแบบสองทิศทางกับ โพรเซส SMTP ฝ่ายรับ จากนั้นคำสั่ง SMTP จากฝ่ายส่งจะถูกส่งให้ฝ่ายรับ ฝ่ายรับก็จะส่งคำตอบสำหรับคำสั่งนั้นกลับไปหาฝ่ายส่งดังรูปที่ 2.2



รูปที่ 2.2 โมเดลของ SMTP

SMTP มีการทำงานที่สามารถเข้าใจได้ง่าย การสื่อสารระหว่างฝ่ายส่งกับฝ่ายรับจะประกอบด้วยข้อความที่อ่านเข้าใจได้ไม่ยาก ในขั้นต้นหลังจากฝ่ายส่งสร้างการเชื่อมต่อกับฝ่ายรับ ฝ่ายส่งจะคอยจนฝ่ายรับส่งข้อความ *220 READY FOR MAIL* กลับมา เมื่อรับข้อความ *220* แล้วฝ่ายส่งจะส่งคำสั่ง *HELO*<sup>1</sup> การจบหนึ่งบรรทัดคือจบหนึ่งคำสั่ง ฝ่ายรับจะตอบกลับมาโดยการแสดงตัวเอง เมื่อการสื่อสารได้เริ่มขึ้น ฝ่ายส่งสามารถส่งจดหมายเพียงฉบับเดียว หรือ หลายๆ ฉบับ หยุดการเชื่อมโยง หรือ ร้องขอให้ฝ่ายรับแลกเปลี่ยนหน้าที่ของฝ่ายส่ง และ ฝ่ายรับ ดังนั้นจดหมายจะสามารถไหลในทิศทางที่ตรงกันข้ามได้ ฝ่ายรับจะต้องยืนยันการได้รับจดหมายแต่ละฉบับ และยังสามารถยกเลิกการเชื่อมโยงทั้งหมด หรือ ยกเลิกการส่งจดหมายปัจจุบันได้

การส่งจดหมายจะเริ่มด้วยคำสั่ง *MAIL FROM:* ที่จะให้ฝ่ายส่งแสดงอีเมลแอดเดรสของผู้ส่งจดหมาย ฝ่ายรับจะเตรียมโครงสร้างข้อมูลเพื่อจะรับจดหมาย และ ตอบคำสั่ง *MAIL* โดยส่งคำตอบรับ *250* คำตอบรับ *250* หมายความว่าทุกอย่างเรียบร้อย คำตอบรับเต็มๆ ประกอบด้วยข้อความ *250 OK* เช่นเดียวกับโปรโตคอลประยุกต์อื่นๆ โปรแกรมที่ใช้ SMTP จะอ่านคำสั่งย่อๆ และ เลขสามหลักที่จุดเริ่มต้นของบรรทัดเป็นสำคัญ ข้อความที่เหลือเจตนาจะช่วยในการดีบั๊ก (Debug) ซอฟต์แวร์เท่านั้น

หลังจากทำคำสั่ง *MAIL* สำเร็จ ฝ่ายส่งจะออกคำสั่ง *RCPT* มาเป็นชุดๆ ซึ่งแต่ละชุดจะแสดงถึงผู้รับของจดหมายฉบับนั้น ฝ่ายรับต้องตอบรับแต่ละคำสั่ง *RCPT* โดยส่ง *250 OK* หรือ โดยส่งข้อความแสดงความผิดพลาด *550 No such user here* ในกรณีที่ไม่มีผู้รับที่ระบุ

<sup>1</sup> *HELO* ย่อมาจาก "hello" ใช้งานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากคำสั่ง *RCPT* ทั้งหมด ฝ่ายส่งจะออกคำสั่ง *DATA* ฝ่ายรับจะตอบด้วยข้อความ *354 Start mail input* และ กำหนดลำดับของตัวอักษรที่ใช้แสดงว่าหมดเนื้อความของจดหมาย ซึ่งประกอบด้วยตัวอักษรห้าตัวคือ Carriage Return, Line Feed, Period, Carriage Return และ Line Feed<sup>2</sup>

สำหรับรายละเอียดของคำสั่งและคำตอบรับทั้งหมดของโปรโตคอล SMTP สามารถดูได้จากภาคผนวก ข.

ตัวอย่างต่อไปนี้แสดงการส่งจดหมายด้วย SMTP สมมติว่าผู้ใช้ชื่อ *Smith* ที่ Host *Alpha.EDU* ส่งจดหมายให้ผู้ใช้ชื่อ *Jones Green* และ *Brown* ที่ host *Beta.GOV* ซอฟต์แวร์ SMTP ฝ่ายส่งบนแม่ข่าย *Alpha.EDU* จะติดต่อกับซอฟต์แวร์ SMTP ฝ่ายรับบนแม่ข่าย *Beta.GOV* และ เริ่มการส่งแสดงได้ดังนี้

```
R: 220 Beta.GOV Simple Mail Transfer Service Ready
S: HELO Alpha.EDU
R: 250 Beta.GOV

S: MAIL FROM:<Smith@Alpha.EDU>
R: 250 OK

S: RCPT TO:<Jones@Beta.GOV>
R: 250 OK

S: RCPT TO:<Green@Beta.GOV>
R: 550 No such user here

S: RCPT TO:<Brown@Beta.GOV>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CR><LF>.<CR><LF>
S: ...sends body of mail message...
S: ...continues for as many lines as message contains
S: .
R: 250 OK

S: QUIT
R: 221 Beta.GOV Service closing transmission channel
```

ในตัวอย่างนี้ฝ่ายรับไม่ยอมรับชื่อผู้รับที่ชื่อ *Green* เพราะไม่ได้เป็นทั้งผู้ใช้ หรือ เมลลิงลิสต์ SMTP ไม่ได้กำหนดรายละเอียดของวิธีที่ฝ่ายส่งจะจัดการกับข้อผิดพลาดเช่นนี้ ซึ่งฝ่ายส่งจะต้องตัดสินใจเอง แม้ว่าฝ่ายส่งจะสามารถยกเลิกการส่งได้ทั้งหมดถ้ามีความผิดพลาดเกิดขึ้น แต่ฝ่ายส่งส่วนใหญ่จะไม่ทำ โดยจะยังส่งข้อมูลให้กับผู้รับที่ถูกต้องทั้งหมด แล้วจึงรายงานปัญหาให้แก่ผู้

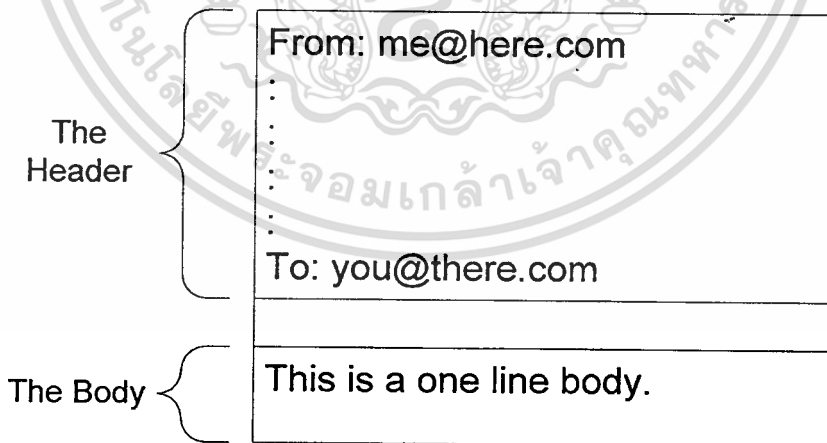
<sup>2</sup>SMTP ไม่อนุญาตให้ body ของจดหมายมีบรรทัดที่มีเพียงจุดๆ เดียว ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งจดหมาย ตามปกติฝ่ายส่งจะรายงานความผิดพลาดโดยใช้จดหมายแสดงความผิดพลาด ซึ่งจะมีสรุปความผิดพลาดรวมทั้ง Header ของจดหมายที่ทำให้เกิดปัญหา

เมื่อฝ่ายส่งสิ้นสุดการส่งจดหมายทั้งหมดที่มีสำหรับปลายทางนั้น ฝ่ายส่งอาจจะออกคำสั่ง *QUIT* อีกฝ่ายจะตอบด้วยคำสั่ง *221* ซึ่งหมายความว่า จะหยุดการติดต่อ แล้วทั้งสองฝ่ายก็จะตัดการเชื่อมต่อลง

## 2.4 รูปแบบของไปรษณีย์อิเล็กทรอนิกส์

รูปแบบของไปรษณีย์อิเล็กทรอนิกส์จะถูกแบ่งเป็นสองส่วนคือ Header และ Body ที่แยกกันด้วยบรรทัดว่างหนึ่งบรรทัดดังรูปที่ 2.3 มาตรฐาน TCP/IP สำหรับรูปแบบของไปรษณีย์อิเล็กทรอนิกส์ [7] กำหนดรูปแบบที่แน่นอนของ Header รวมทั้งความหมายของแต่ละ Header และปล่อยให้รูปแบบของ Body เป็นไปตามแต่ผู้ส่ง มาตรฐานกำหนดเป็นการเฉพาะให้ Header บรรจุหัวข้อที่สามารถอ่านได้โดยแบ่งเป็นบรรทัดที่ประกอบด้วย Keyword ตามด้วย Colon ตามด้วยค่า บาง Keyword จำเป็นต้องมี บาง Keyword ก็เป็น Option และส่วนที่เหลือจะไม่ถูกตีความ ตัวอย่างเช่น Header ต้องมีบรรทัดที่ระบุจุดหมายปลายทาง บรรทัดที่ขึ้นต้นด้วย *To:* และมีอีเมลแอดเดรสของผู้รับบนส่วนที่เหลือของบรรทัด บรรทัดที่ขึ้นต้นด้วย *From:* จะแสดงอีเมลแอดเดรสของผู้ส่ง ผู้ส่งอาจจะเลือกกำหนดอีเมลแอดเดรสที่จะให้ตอบกลับไปได้ นอกเหนือจากอีเมลแอดเดรสปกติของผู้ส่ง ถ้ามีบรรทัดที่ขึ้นต้นด้วย *Reply-To:* จะกำหนดอีเมลแอดเดรสสำหรับตอบกลับ ถ้าไม่มีบรรทัดนี้อยู่ ผู้รับจะใช้ข้อมูลจากบรรทัด *From:* เป็นอีเมลแอดเดรสที่จะตอบกลับ



รูปที่ 2.3 ไปรษณีย์อิเล็กทรอนิกส์ประกอบด้วย Header และ Body

รูปแบบของไปรษณีย์อิเล็กทรอนิกส์ถูกกำหนดให้ง่ายที่จะประมวลผล และง่ายที่จะส่งผ่านเครื่องที่ต่างชนิดกัน การเก็บ Header เป็นรูปแบบอย่างง่าย ๆ นี้ทำให้ใช้บนระบบได้กว้างขวาง

การจำกัดให้เนื้อความเป็นข้อความที่อ่านได้จะหลีกเลี่ยงปัญหาของการเลือกมาตรฐานที่ใช้แทนรหัสฐานสอง (Binary) และการแปลงระหว่างรหัสมาตรฐานกับรหัสของเครื่องของท้องถิ่น

## 2.5 เมลลิงลิสต์

เมลลิงลิสต์ (Mailing List) หรือ บัญชีจำหน่าย คือการประยุกต์ใช้ไประณีย์อิเล็กทรอนิกส์รูปแบบหนึ่ง โดยใช้ประโยชน์จากความสามารถในการส่งอีเมลเพียงฉบับเดียวถึงผู้รับหลายๆ คนได้โดยง่าย เพียงจำหน่ายถึงผู้รับเหล่านั้นหรืออาจจะแทนด้วยการทำ Alias (เช่นเมื่อใช้ sendmail เป็น Mail Transport Agent) ซึ่งเป็นการแทนรายชื่อของผู้รับเหล่านั้นด้วยชื่อ Alias เพียงชื่อเดียว Mail Transport Agent ก็จะทำสำเนาอีเมลฉบับนั้นส่งให้ผู้รับทุกคนโดยอัตโนมัติ ด้วยเหตุนี้จึงมีการนำเมลลิงลิสต์มาประยุกต์ใช้ในการประชุม ปรึกษาหารือ แลกเปลี่ยนความคิดเห็น หรือเพื่อแจ้งข่าวสารถึงกลุ่มสมาชิก

## 2.6 การจัดการการเมลลิงลิสต์

การจัดการเมลลิงลิสต์นั้นถ้าสมาชิกที่ใช้ค่อนข้างคงที่ไม่มีการเปลี่ยนแปลงบ่อยนัก ก็อาจไม่จำเป็นต้องใช้ซอฟต์แวร์ช่วยในการจัดการ ถ้าเมลลิงลิสต์นั้นต้องถูกกั้นกรองก่อน (Moderated หรือตรวจสอบความเหมาะสมของเนื้อหาในแต่ละฉบับก่อนที่จะกระจายให้แก่สมาชิก) ก็อาจใช้เพียงแค่โปรแกรมจัดการอีเมล (Mail User Agent) ทั่วๆ ไปเช่น Pine mh หรือ Eudora ก็เพียงพอต่อการดูแลเมลลิงลิสต์แล้ว เมื่อมีจดหมายมาถึง ก็ใช้โปรแกรมนั้นทำการส่งต่อไปให้สมาชิกในเมลลิงลิสต์ สิ่งที่ต้องทำสำหรับแนวทางนี้คือ ถ้าโปรแกรมที่ใช้มีฟังก์ชัน "Bounce" ก็ควรจะใช้ฟังก์ชันนี้แทน "Forward" เพราะมันจะรักษา Header "From:" ซึ่งแสดงว่าใครเป็นคนส่งต้นฉบับของจดหมาย และแสดงอีเมลแอดเดรสของเจ้าของเมลลิงลิสต์ที่ "Resent-From:" ซึ่งแบบนี้จะเหมาะสมกว่า นอกจากนี้ยังควรใส่รายชื่อของอีเมลแอดเดรสที่ "Bcc:" แทนที่จะเป็น "To:" เพื่อว่าถ้าสมาชิกตอบจดหมายฉบับนั้นมันจะได้ส่งกลับมาที่เจ้าของเมลลิงลิสต์ไม่ใช่ส่งให้เมลลิงลิสต์ (โดยไม่ผ่านการกั้นกรองจากเจ้าของเมลลิงลิสต์)

นอกจากนี้ยังมีวิธีที่สามารถติดตั้งระบบอัตโนมัติเพื่อการส่งอีเมลกลับให้สมาชิกได้ ถ้าใช้ระบบยูนิคซ์ที่รันโปรแกรม sendmail โดยใช้ฟังก์ชัน "Alias" ของ sendmail สามารถสร้างเมลลิงลิสต์ขึ้นโดยแก้ไขแฟ้ม /etc/aliases ซึ่งมีรูปแบบดังต่อไปนี้

```
alias: ผู้รับ[, ผู้รับ,...]
```

alias คือชื่อที่จะใช้ส่งจดหมายให้ ส่วน ผู้รับ คือรายชื่อผู้รับที่จดหมายจะถูกส่งไปถึงซึ่งสามารถเป็นได้ทั้งชื่อผู้ใช้ alias หรือ อีเมลแอดเดรส ตัวอย่างเช่น

```
admin: bob,jim,phill
bob: \bob,/u/bob/admin/maillog
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างชื่อ admin เป็นชื่อของเมลลิงลิสต์ เพราะว่าสามารถขยายเป็นมากกว่าหนึ่งผู้รับ เช่นเดียวกับ bob เพราะ bob ก็สามารถขยายเป็นสองผู้รับได้ และเนื่องจาก bob เป็นสมาชิกของเมลลิงลิสต์ admin ดังนั้นจดหมายที่ส่งถึงเมลลิงลิสต์ admin ก็จะถูกกระจาย โดย sendmail ถึงผู้รับดังรายชื่อต่อไปนี้

```
jim, phil, \bob, /u/bob/admin/maillog
```

ซึ่งผู้รับ \bob จะถูกส่งถึงโดยไม่มีการขยาย alias อีกเพราะว่าระบุไว้ด้วย backslash (“\”) และสุดท้ายจดหมายจะถูกผนวก (Append) เข้ากับแฟ้มข้อมูล /u/bob/admin/maillog

การจัดการเมลลิงลิสต์ด้วยวิธีนี้เหมาะกับเมลลิงลิสต์ที่มีขนาดเล็กและไม่เปลี่ยนแปลงสมาชิกบ่อยๆ นอกจากนี้ผู้ที่มีสิทธิแก้ไขแฟ้ม aliases ได้คือ root เท่านั้น ซึ่งถ้าจะให้ผู้ดูแลเมลลิงลิสต์ที่ไม่ใช่ root แก้ไขได้อาจต้องแก้ไขให้เพิ่ม aliases นี้แก้ไขโดยคนอื่นได้ ซึ่งอาจเกิดปัญหาด้านความปลอดภัยขึ้นได้ ดังนั้นผู้ดูแลระบบอาจจะติดตั้ง Alias พิเศษชนิดหนึ่งให้กับลิสต์ ผู้ดูแลเมลลิงลิสต์ก็จะสามารถดูแลลิสต์ได้โดยง่ายเพียงแต่ทำการแก้ไขไฟล์ซึ่ง Alias นั้นชื่ออยู่ เมื่อมีเมลส่งมาที่ Alias นี้ เมลฉบับนั้นก็จะกระจายถึงทั้งลิสต์ ซึ่งมีรูปแบบในแฟ้ม aliases ดังนี้

```
listname: :include:/path
```

ซึ่ง /path คือ Pathname เต็มของแฟ้มข้อมูลที่บรรจุรายชื่อของผู้รับ โดยแต่ละบรรทัดของแฟ้มนี้จะถือว่าเป็นรายชื่อของผู้รับหนึ่งผู้รับหรือถ้ามากกว่าต้องแบ่งด้วย Comma (“;”)

อย่างไรก็ตามแนวทางนี้อาจเกิดปัญหาได้มากเช่น Mailing Loops และยังไม่สามารถควบคุมว่าใครบ้างที่สามารถส่งเมลให้ลิสต์ ดังนั้นเมลลิงลิสต์ที่ต้องการการจัดการและการกระจายจดหมายแบบอัตโนมัติมักเลือกใช้ซอฟต์แวร์จัดการเมลลิงลิสต์

## 2.7 ซอฟต์แวร์จัดการเมลลิงลิสต์ (Mailing List Manager)

ซอฟต์แวร์จัดการเมลลิงลิสต์ คือซอฟต์แวร์ที่ควบคุมระบบสมาชิกของเมลลิงลิสต์ ไม่ว่าจะเป็นการสมัครเป็นสมาชิกของเมลลิงลิสต์หรือการบอกเลิกสมาชิก และนอกจากนี้ยังมักจะควบคุมการกระจายจดหมายให้แก่สมาชิกของเมลลิงลิสต์อีกด้วย จากการศึกษาพบว่า มีซอฟต์แวร์จัดการเมลลิงลิสต์อยู่หลายตัวดังต่อไปนี้

1. **Almanac** ถูกพัฒนาขึ้นที่ Oregon State University, Extension Service โดยภาษา C มีบริการส่งแฟ้มข้อมูลผ่านทางจดหมาย และยังมีคุณสมบัติของรูปแบบในการส่งคำร้องขอมาก เช่น สามารถส่งคำร้องขอเป็นภาษาฝรั่งเศสได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **BMW** ย่อมาจาก The Black Marble Wombat เขียนด้วย Perl สามารถใช้ได้ฟรีภายใต้ GNU license
3. **IDG** ย่อมาจาก Internet Discussion Group สามารถทำงานได้โดยใช้ Account ธรรมดาเพียง Account เดียว
4. **ListProc, free version (v. 6.0c)** เคยถูกเรียกว่า Unix Listserv. เพราะว่าเลียนแบบการทำงานของ LISTSERV ซึ่งในขณะนั้นยังไม่สามารถทำงานบน Unix ได้ ต่อมาจึงเปลี่ยนชื่อเพื่อหลีกเลี่ยงความสับสน
5. **CREN ListProc (v. 7 ขึ้นไป)** CREN, the Corporation for Research and Educational Networking ได้ซื้อลิขสิทธิ์ ListProc ไปเพื่อพัฒนาต่อและจำหน่ายในเชิงพาณิชย์ ทำให้สนับสนุนหลากหลายแพลตฟอร์มขึ้น
6. **LISTSERV** บางทีก็ถูกเรียกว่า BITNET LISTSERV แต่ปัจจุบันนี้ไม่ได้ทำงานแค่บน BITNET อีกต่อไปแล้ว และน่าจะเป็นซอฟต์แวร์จัดการเมลลิ่งลิสต์ ที่มีลักษณะสำคัญเทียบพร้อมที่สุด จัดจำหน่ายโดย L-Soft International
7. **Mailbase** เป็นที่นิยมมากในอังกฤษ ทำงานบน SunOS 4.1.1 และ 4.1.3 และ Solaris 2.3 และยังต้องมี Ingres 6.3 ขึ้นไป และ Perl
8. **MAISER** เป็นส่วนหนึ่งของ Mercury Mail Transport System ทำงานบน Netware Server
9. **Majordomo** เป็นซอฟต์แวร์จัดการเมลลิ่งลิสต์ ที่ได้รับการพัฒนาอย่างต่อเนื่องและได้รับความนิยมเพิ่มขึ้นอย่างรวดเร็ว เขียนด้วย Perl และใช้ได้ฟรี
10. **Mreply** สามารถติดตั้งได้โดยไม่ต้องเป็น root มีความยืดหยุ่นสูง
11. **MXSERV (MX/MLF, เป็นส่วนหนึ่งของ Message Exchange system)** MX (Message Exchange) เป็นระบบจดหมายฟรีบน VMS ประกอบด้วย Delivery Agent หลายตัว หนึ่งในนั้นคือ MLF (Mailing List/File Server) ซึ่งทำหน้าที่จัดการเมลลิ่งลิสต์
12. **SmartList** ทำงานโดยอาศัย Procmail ซึ่งเป็น Local Delivery Agent สำหรับยูนิกซ์ ลักษณะเด่นคือทำให้ผู้ใช้รู้สึกเหมือนกำลังติดต่อกับคนจริงๆ มากกว่ากับโปรแกรม เนื่องจากมันรับคำสั่งได้ทุกรูปแบบ รวมทั้งข้อความธรรมดา SmartList ไม่ทำให้ระบบทำงานหนักแต่ยังมีสมรรถนะในการส่งสูง เหมาะกับเมลลิ่งลิสต์ขนาดใหญ่ๆ นอกจากนี้ยังจัดการกับ Bounced Mail ได้ดีกว่าซอฟต์แวร์จัดการเมลลิ่งลิสต์ตัวอื่น
13. **Smof Listserv for DOS/KA9Q** ทำงานบน DOS ที่มี KA9Q
14. **TULP** เกิดขึ้นมาคล้าย ListProc สมัยที่ LISTSERV ยังทำงานได้แค่บน VM/CMS เพื่อทำงานแบบ LISTSERV บนยูนิกซ์ จึงเลียนแบบการทำงานของ LISTSERV โดยเฉพาะคุณสมบัติพื้นฐานทุกอย่าง แต่มีคุณสมบัติเรียบง่ายและไม่หรูหรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8 สมรรถภาพในการกระจายจดหมายของซอฟต์แวร์จัดการเมลถึงลิสต์

ประเด็นหลักในการกระจายจดหมายให้แก่เมลถึงลิสต์ขนาดใหญ่คือ ความเร็วของซอฟต์แวร์จัดการเมลถึงลิสต์ในการส่งจดหมายออก ซอฟต์แวร์จัดการเมลถึงลิสต์ส่วนใหญ่จะปล่อยให้เป็นที่ของ Mail Transport Agent (มักจะเป็น sendmail ถ้าใช้ยูนิกซ์) ในการตัดสินใจจัดการและวางเส้นทางในการส่ง (Routing) แต่ซอฟต์แวร์จัดการเมลถึงลิสต์บางตัวโดยเฉพาะอย่างยิ่ง ListProc LISTSERV และ SmartList จะจัดการภาระในการกระจายจดหมายด้วยตัวเอง

Majordomo เป็นตัวอย่างที่ดีของกรวยกให้เป็นหน้าที่ของ sendmail ทั้งหมด เมื่อ Majordomo ต้องการกระจายจดหมายให้กับเมลถึงลิสต์ก็จะส่งจดหมายฉบับนั้นพร้อมด้วยอีเมลแอ็คเตอรของทั้งเมลถึงลิสต์ให้โดยการเรียก sendmail ขึ้นมาทำงานเพียงหนึ่งโปรเซส จากนั้น sendmail ก็จะทำหน้าที่ของมันในการทำให้สามารถส่งได้เร็วที่สุด โดยการเรียงลำดับการส่งตาม MX Record และเริ่มเชื่อมต่อกับเครื่องที่มีอีเมลแอ็คเตอรตาม MX Record เดียวกัน ทำให้หากสมาชิกจำนวนมากแม้จะอยู่ใกล้เคียงกัน แต่ถ้ามี MX Record ต่างกัน อาจทำให้เกิดความล่าช้าในการส่งเป็นชั่วโมงระหว่างสมาชิกคนแรกกับคนสุดท้าย ซึ่งขึ้นอยู่กับความเร็วของเครื่องของสมาชิกมากกว่า ดังนั้นเมลถึงลิสต์ที่มีขนาดใหญ่และมีปริมาณจดหมายหนาแน่นจะทำให้สมาชิกที่ได้รับการส่งจดหมายให้ในลำดับท้ายๆ อาจได้รับและตอบจดหมายใหม่ที่สมาชิกคนอื่นๆ ได้โต้ตอบกันไปแล้วเมื่อหลายชั่วโมงก่อน การสนทนาก็อาจเกิดความไม่ต่อเนื่องกันได้

Listproc จะเพิ่มความเร็วของกระบวนการกระจายจดหมายให้กับเมลถึงลิสต์ขนาดใหญ่ โดยแบ่งอีเมลแอ็คเตอรของสมาชิกเรียงตามโดเมนออกเป็นกลุ่มๆ ตามขนาดที่สามารถกำหนดได้จากนั้นก็ส่งให้ Mail Transport Agent ผ่านทาง Port 25 ด้วย SMTP ทำให้การส่งมักจะเร็วขึ้นเนื่องจากจะมีการส่งแบบขนานกันไปขึ้นอยู่กับ Mail Transport Agent ว่าถูกกำหนดให้ทำงานอย่างไร อย่างไรก็ตามอาจเป็นการเพิ่มภาระให้กับเครื่องที่ทำการส่งเนื่องจาก Mail Transport Agent หลายๆ โปรเซสจะทำงานพร้อมๆ กันไป

Smartlist ใช้วิธีคล้าย Listproc คือแบ่งอีเมลแอ็คเตอรของสมาชิกออกเป็นกลุ่มๆ ก่อนที่จะส่งให้แก่ Mail Transport Agent (ผ่านทาง Command Line ไม่ใช่ Port 25) แต่พัฒนาขึ้นไปอีกขั้นหนึ่ง คือแทนที่จะแค่กำหนดขนาดของแต่ละกลุ่มได้ Smartlist จะให้ขอมผู้ดูแลระบบกำหนดขนาดสูงสุดและต่ำสุดไว้ และให้ Smartlist ตัดสินใจขนาดที่เหมาะสมเอง เช่นพยายามให้อีเมลแอ็คเตอรที่อยู่ในโดเมนเดียวกันอยู่ร่วมกันเป็นกลุ่มเดียว นอกจากนี้ผู้ดูแลระบบยังสามารถกำหนดจำนวนโปรเซสสูงสุดของ Mail Transport Agent ที่จะทำงานพร้อมกัน ทำให้สามารถควบคุมความเร็วในการส่งกับภาระของระบบได้โดยตรง

LISTSERV ใช้โพรโทคอลพิเศษที่เรียกว่า “Distribute” [9] ที่อาศัยความได้เปรียบจากความร่วมมือกันของระบบที่ใช้ LISTSERV ทั่วโลก โดยในขั้นแรก LISTSERV จะส่งจดหมายให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

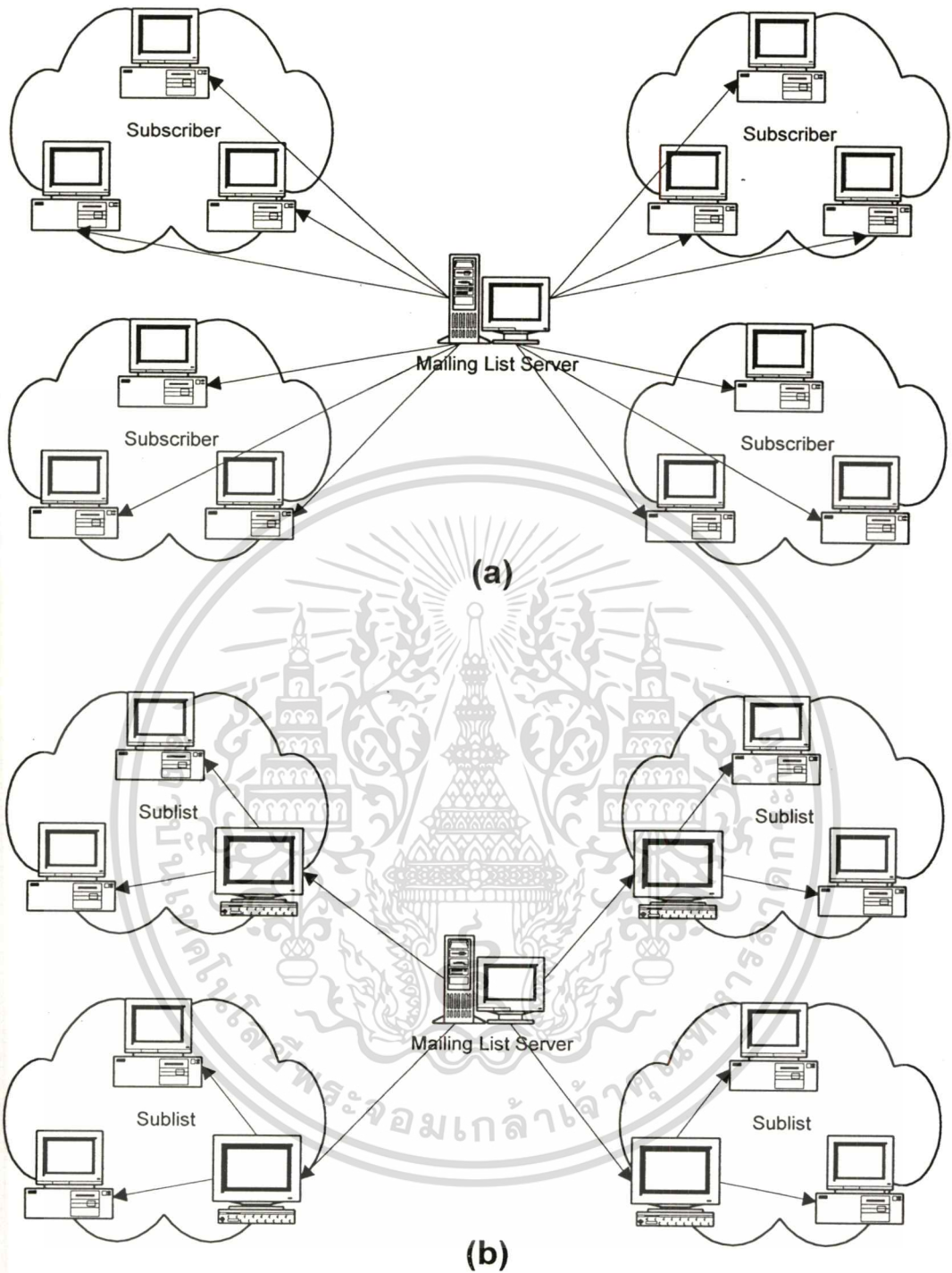
สมาชิกที่อยู่ในเครือข่ายเดียวกับมันก่อน โดยเชื่อมต่อกับ Mail Transport Agent ผ่าน Port 25 เช่นเดียวกับ ListProc จากนั้นก็จะทำการค้นหารายชื่ออีเมลแอดเดรสของสมาชิกในเมลลิงลิสต์ที่อยู่ในเครือข่ายอื่นที่ใช้ LISTSERV เหมือนกัน จากนั้นจึงทำการส่งอีเมลแอดเดรสเหล่านั้นพร้อมด้วยจดหมายเพียงฉบับเดียวให้กับ LISTSERV Core Site ที่ใกล้เคียงให้รับภาระในการส่งต่อไป และสุดท้าย LISTSERV จะจัดการกับอีเมลแอดเดรสที่เหลือ (ซึ่งไม่สามารถหา LISTSERV ที่อยู่ใกล้เคียงได้) โดยการส่งเป็นกลุ่มเช่นเดียวกับ ListProc ยิ่งเมื่อ LISTSERV สามารถหาสมาชิกที่อยู่ใกล้เคียงกับ LISTSERV ที่อื่นๆ ได้มาก และยิ่งสมาชิกเหล่านั้นมีการเชื่อมต่อที่ช้าและต้องเสียค่าใช้จ่ายสูงวิธีนี้จะทำให้มีประสิทธิภาพในการกระจายจดหมายสูงมาก ตัวอย่างเช่น จดหมายฉบับหนึ่งจากอเมริกาถึงสมาชิกนับพันในยุโรปที่อยู่ในแม่ข่ายต่างๆ กัน จะข้ามมหาสมุทรแอตแลนติกไปเพียงลำเนาเดียวแล้วจึงแตกกระจายออก ดังนั้นโพรโทคอล Distribute จึงสามารถลดภาระทั้งของเครื่องและเครือข่ายได้ และยังทำให้มีความเร็วในการส่งสูงเพราะว่าใช้เครื่องหลายๆ เครื่องจะทำการส่งไปพร้อมๆ กัน แต่อย่างไรก็ตามโพรโทคอลนี้จะไม่เป็นประโยชน์สำหรับเมลลิงลิสต์ที่มีสมาชิกที่อยู่ไกลๆ น้อย และนอกจากนี้ยังเป็นผลให้เครื่องที่บริการแต่ละเครื่องต้องรับภาระในการส่งจดหมายของเครื่องอื่น ซึ่งอาจเป็นการเพิ่มภาระของเครื่องนั้นๆ

แต่โพรโทคอล Distribute นี้ไม่เป็นที่นิยมใช้เนื่องจาก มันสามารถทำงานได้เฉพาะกับ LISTSERV เพื่อใช้งานกับ BITNET ซึ่งไม่เหมาะจะนำมาใช้กับเมลลิงลิสต์ของอินเทอร์เน็ตเนื่องจากโพรโทคอลนี้จะคำนวณหาเส้นทางจากตารางที่คงที่ ซึ่งเป็นวิธีที่เหมาะสมกับ BITNET ซึ่งเป็นเครือข่ายที่ไม่ค่อยมีการเปลี่ยนแปลง ต่างจากอินเทอร์เน็ตที่มักจะเปลี่ยนแปลงบ่อยๆ ดังนั้นหากต้องการการกระจายจดหมายที่มีประสิทธิภาพแบบเดียวกับโพรโทคอล Distribute ก็ควรสร้างเมลลิงลิสต์ย่อยในจุดที่เหมาะสม เช่น ระหว่างเครือข่ายที่ช้าและต้องเสียค่าใช้จ่ายสูง และในใจกลางของกลุ่มสมาชิกจำนวนมาก

## 2.9 การแก้ปัญหาความซ้ำซ้อนของจดหมายจากเมลลิงลิสต์โดยใช้ลิสต์ย่อย

### 2.9.1 เมลลิงลิสต์ย่อย

เนื่องจากอินเทอร์เน็ตเมลลิงลิสต์ใดๆ จะส่งจดหมายถึงสมาชิกทุกคนโดยไม่สนใจความใกล้เคียงกันของเครือข่ายของสมาชิก ด้วยเหตุนี้จึงเกิดความซ้ำซ้อนของข้อมูลที่ส่งมาหากมีสมาชิกของเมลลิงลิสต์นั้นอยู่ในเครือข่ายเดียวกันหรือเครือข่ายใกล้เคียงกันมากกว่าหนึ่งคน ดังนั้นหากต้องการลดความซ้ำซ้อนนี้จึงควรรวมกลุ่มสมาชิกของเมลลิงลิสต์นั้นเข้าตามความใกล้เคียงกันของเครือข่าย ตั้งเป็นเมลลิงลิสต์ย่อยขึ้นมาเพื่อรับจดหมายจากเมลลิงลิสต์หลักเพียงลำเนาเดียวต่อฉบับแล้วจึงกระจายต่อให้สมาชิกภายในลิสต์ย่อยในภายหลัง ดังรูปที่ 2.4



รูปที่ 2.4 แสดงโครงสร้างของเมลลิ่งลิสต์ (a) แบบปกติ (b) แบบมีลิสต์ย่อย

จากรูปจะเห็นได้ว่านอกจากจะเป็นการลดความซ้ำซ้อนของจดหมายที่ต้องส่งระหว่างเครือข่ายลงแล้ว ยังเป็นการลดจำนวนสมาชิกของเมลลิ่งลิสต์หลักลง ซึ่งจะเกิดประโยชน์คือ เป็นการกระจายภาระในการจัดการเมลลิ่งลิสต์ขนาดใหญ่ออกไปให้ลิสต์ย่อย ทำให้ภาระในการจัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมลลิงลิสต์ของเมลลิงลิสต์เซิร์ฟเวอร์หลักลดลง ช่วยให้ประสิทธิภาพในการกระจายจดหมายดีขึ้น ซึ่งเป็นผลดีโดยรวมกับทั้งเมลลิงลิสต์

## 2.9.2 การทำงานของซอฟต์แวร์จัดการเมลลิงลิสต์ที่เป็นอุปสรรคต่อเมลลิงลิสต์ย่อย

หน้าที่พื้นฐานที่สำคัญอย่างหนึ่งของซอฟต์แวร์จัดการเมลลิงลิสต์คือ เตรียมจดหมายเพื่อให้พร้อมสำหรับการแจกจ่ายในวงกว้าง และป้องกันสมาชิกของเมลลิงลิสต์จากบุคคลอื่นที่พยายามจะปลอมตัวเป็นสมาชิก

เมื่อจดหมายถูกส่งต่อให้เมลลิงลิสต์ Header บางส่วนจะถูกตัดออก ตัวอย่างเช่น Header “Return-Receipt-To:” ซึ่งถ้ากระจายสู่เมลลิงลิสต์อาจสามารถส่งผลให้เกิดจดหมายตอบรับอัตโนมัติกลับสู่ระบบของผู้ส่งอย่างท่วมท้น แนวทางในการกำจัด Header เหล่านี้มีสองแนวทางคือ กำจัด Header ที่รู้ว่าจะเป็นอันตราย และอนุญาตเฉพาะ Header ที่รู้ว่าจะปลอดภัย Majordomo ใช้แนวทางแรกคือตัด Header ต้องห้ามด้วยรูทีน “Resend” ของมัน ในขณะที่ Listproc ใช้แนวทางตรงกันข้ามคืออนุญาตเฉพาะ Header ที่มันรู้จักและตัดที่เหลือทิ้ง เกณฑ์ในการพิจารณา Header ของซอฟต์แวร์ทั้งสองสามารถแก้ไขได้โดยผู้ดูแลระบบ

บางครั้งซอฟต์แวร์จัดการเมลลิงลิสต์ก็ต้องการความแน่ใจว่า บุคคลที่ส่งจดหมายมาเป็นคนคนนั้นจริงๆ เนื่องจากการปลอมเมลเป็นเรื่องง่าย ดังนั้น Header “From:” ก็ไม่สามารถเชื่อถือได้แม้มันจะเป็นหลักฐานเพียงอย่างเดียวที่ซอฟต์แวร์จัดการเมลลิงลิสต์ส่วนใหญ่ต้องการ ยกเว้น

1. ListProc 6.0c ใช้บรรทัด SMTP “From ” แต่สามารถแก้ไขให้ใช้ “From:” เหมือนตัวอื่นได้
2. Majordomo ขอมให้ผู้ส่งเปลี่ยนหลักฐานที่ใช้แสดงตัวได้โดยใช้ Header “Reply-To:” เพราะถือว่าการปลอมเมลเป็นเรื่องง่าย ทำไม่ไม่ทำให้การแสดงตัวอย่างถูกต้องเป็นเรื่องที่ง่ายกว่า ซึ่งมีข้อดีคือช่วยทำให้สมาชิกที่ต้องการย้ายอีเมลแอดเดรสสามารถจัดการได้ด้วยตัวเองโดยไม่ต้องขอความช่วยเหลือจากเจ้าของเมลลิงลิสต์
3. SmartList จะอ่าน Header ที่เกี่ยวกับผู้ส่งทั้งหมดรวมทั้ง “From ” “From:” “Sender:” “Reply-To:” และ Signature Text ถ้ามี แล้วจึงเลือกใช้แล้วแต่สถานการณ์

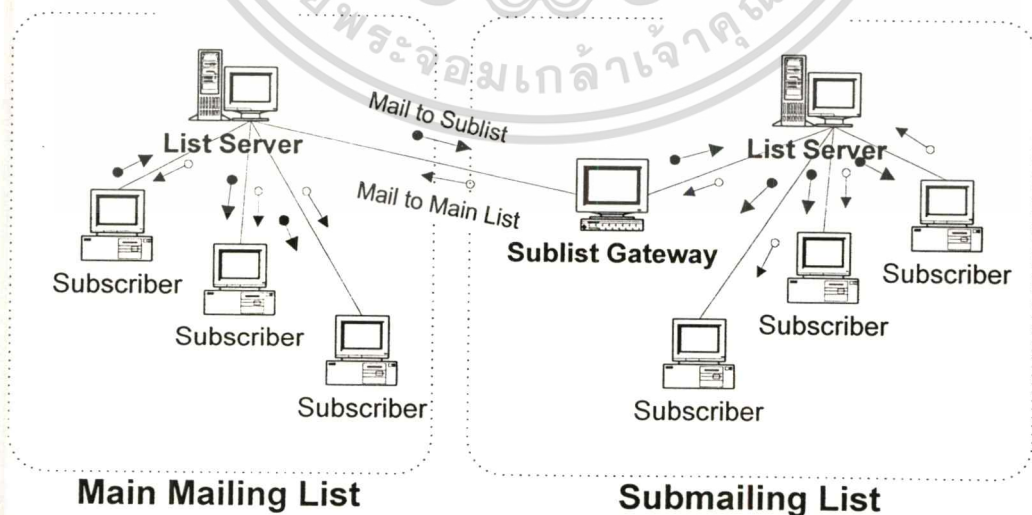
นอกจากนี้เพื่อป้องกันไม่ให้เกิดการวนซ้ำ (Mailing Loop) ซึ่งจะเกิดเมื่อเซิร์ฟเวอร์สองเครื่องต่างถูกกำหนดให้ Forward จดหมายให้แก่กัน โดยไม่รู้ว่ามีอีกฝั่งก็จะ Forward จดหมายฉบับนั้นกลับมาให้ การป้องกันทำได้หลายวิธี อาทิเช่น ซอฟต์แวร์จัดการเมลลิงลิสต์ส่วนใหญ่จะปฏิเสธจดหมายที่มี Header “Precedence” เป็น “Bulk” และจะเติม Precedence “Bulk” นี้ให้กับจดหมายที่ส่งไปยังเมลลิงลิสต์เพื่อป้องกันไม่ให้โปรแกรมจำพวก Vacation (โปรแกรมอัตโนมัติที่ตอบรับจดหมายแทน ใช้เมื่อผู้รับไม่สามารถอยู่ตอบจดหมายได้ทันที) ตอบรับ

### 2.9.3 การสร้างเมลลิงลิสต์ย่อย

ปกติการสร้างเมลลิงลิสต์ย่อยสามารถทำได้อยู่แล้วเช่นเดียวกับการสร้างเมลลิงลิสต์หลัก โดยการสร้าง Mail Alias ขึ้นมาสำหรับกระจายจดหมายให้สมาชิกในเมลลิงลิสต์ย่อย จากนั้นก็ใช้ Alias นี้สมัครเป็นสมาชิกของเมลลิงลิสต์หลัก เมื่อเมลลิงลิสต์หลักกระจายจดหมายมาให้ Alias นี้ก็จะกระจายต่อให้สมาชิกในเมลลิงลิสต์ย่อยต่อไป แต่วิธีนี้ยังมีข้อจำกัดอยู่ เนื่องจากซอฟต์แวร์จัดการเมลลิงลิสต์โดยทั่วไปจะไม่ยอมให้ผู้ที่ไม่ใช่สมาชิกของลิสต์ส่งข้อความดังที่ได้กล่าวมาแล้ว ดังนั้นเมื่อสมาชิกของเมลลิงลิสต์ย่อยต้องการส่งจดหมายโต้ตอบ จดหมายของสมาชิกในลิสต์ย่อยก็จะไม่สามารถกระจายสู่สมาชิกของเมลลิงลิสต์หลักคนอื่นๆ ได้ เนื่องจากพวกเขาไม่ได้มีรายชื่ออยู่ในเมลลิงลิสต์หลัก

และด้วยข้อจำกัดเดียวกันนี้ การจัดการเมลลิงลิสต์ย่อยยังไม่สามารถใช้ซอฟต์แวร์จัดการเมลลิงลิสต์ต่างๆ ไปมาจัดการได้ เนื่องจากมันก็จะป้องกันไม่ให้จดหมายสมาชิกคนอื่นๆ ของเมลลิงลิสต์หลักกระจายสู่สมาชิกในเมลลิงลิสต์ย่อย เพราะว่าสมาชิกของเมลลิงลิสต์หลักไม่ได้เป็นสมาชิกของเมลลิงลิสต์ย่อยที่มันจัดการอยู่จึงไม่มีสิทธิในการส่งจดหมายให้ ดังนั้นการจัดการเมลลิงลิสต์ย่อย ถ้าไม่จัดการด้วยตนเอง (Manual) ก็ต้องกำหนดให้ซอฟต์แวร์ที่ใช้จัดการเมลลิงลิสต์ย่อยยอมให้ใครก็ได้ที่ไม่ใช่สมาชิกส่งจดหมายให้เมลลิงลิสต์ย่อย ซึ่งทำให้ความเป็นส่วนตัวของสมาชิกลดลง และถูกก่อกวนได้ง่ายขึ้น

ดังนั้นจึงต้องสร้างระบบที่ใช้จัดการเมลลิงลิสต์ย่อยขึ้นมาเพื่อแก้ไขข้อจำกัดนี้ โดยทำหน้าที่เสมือนเป็นสมาชิกของทั้งเมลลิงลิสต์หลักและเมลลิงลิสต์ย่อย และคอยถ่ายทอดจดหมายระหว่างเมลลิงลิสต์หลักและเมลลิงลิสต์ย่อย ทำให้เกิดการโต้ตอบกันได้ทั้งสองฝั่งโดยไม่ถูกจำกัดด้วยข้อจำกัดที่กล่าวมา ซึ่งมีรายละเอียดดังรูปที่ 2.5



รูปที่ 2.5 ทิศทางในการถ่ายทอดจดหมาย

จากรูปแสดงทิศทางของจดหมายที่ไหลเวียนอยู่ในระบบจัดการเมลลิงลิสต์ย่อย แสดงให้เห็นว่า Sublist gateway ซึ่งเป็นสมาชิกทั้งเมลลิงลิสต์หลักและเมลลิงลิสต์ย่อย จะได้รับจดหมายจากสมาชิกของทั้งสองเมลลิงลิสต์ เมื่อได้รับจดหมายจากเมลลิงลิสต์ใด ก็จะทำเสมือนเอาจดหมายฉบับนั้นใส่ลงในช่องของใหม่ และจำหน่ายของว่าจดหมายฉบับนั้นส่งมาจาก Sublist gateway เอง แล้วจึงส่งต่อให้กับเมลลิงลิสต์ปลายทางอีกฝั่งหนึ่ง เมื่อเป็นดังนี้เมลลิงลิสต์ปลายทางก็จะยอมรับจดหมายฉบับนั้นส่งให้กับสมาชิกต่อไป แม้ว่าที่จริงแล้วจดหมายฉบับนั้นจะไม่ได้มาจากสมาชิกของเมลลิงลิสต์ปลายทางก็ตาม

ดังนั้นสมาชิกในเมลลิงลิสต์ย่อยก็จะสามารถได้ตอบกับสมาชิกในเมลลิงลิสต์หลักได้ เหมือนกับว่าเป็นสมาชิกของเมลลิงลิสต์เดียวกัน และนอกจากนี้ ยังสามารถนำซอฟต์แวร์จัดการเมลลิงลิสต์ตัวใดก็ได้ มาจัดการเมลลิงลิสต์ย่อย โดยไม่ต้องแก้ไขซอฟต์แวร์จัดการเมลลิงลิสต์นั้น



## บทที่ 3

# การใช้โปรโตคอล MIME ในการแก้ปัญหาของเมลลิงลิสต์

### 3.1 บทนำ

จากปัญหาเรื่องการสลับเปลี่ยนเนื้อหาในการเก็บจดหมายจากเมลลิงลิสต์ที่ซ้ำซ้อนกันในผู้ใช้ที่เป็นสมาชิกของเมลลิงลิสต์เดียวกัน ที่อยู่ภายในเครือข่ายเดียวกันหรือใกล้เคียงกันนั้น แนวทางในการแก้ปัญหาคือการใช้เนื้อหาที่จัดเก็บซ้ำซ้อนกัน สามารถแก้ไขโดยใช้ลักษณะการจัดเก็บเช่นเดียวกับ Usenet กล่าวคือ เก็บจดหมายแต่ละฉบับไว้เพียงก๊อปปี้เดียวสำหรับเครือข่ายที่ใกล้เคียงกัน เมื่อผู้ใช้อ่านจดหมายก็ค่อยจัดการดึงจดหมายที่เก็บไว้มาแสดง จะเห็นได้ว่าการที่จะกระทำเช่นนี้ได้ ตัว Mail User Agent หรือโปรแกรมที่แสดงจดหมายแก่ผู้ใช้จะต้องรู้จักการจัดเก็บแบบนี้ และต้องสามารถดึงมาใช้ได้ ซึ่งคงไม่เหมาะสมถ้าต้องตามแก้ไข User agent ทุกตัวให้ทำเช่นนี้ได้

นอกจากนี้ การแก้ไข Header ของจดหมายต้นฉบับเพื่อให้สามารถส่งต่อไปกับอีกเมลลิงลิสต์หนึ่งได้ เนื่องจากผู้ส่งจดหมายต้นฉบับไม่ได้เป็นสมาชิกของเมลลิงลิสต์ปลายทางที่จะส่งจดหมายต่อให้มัน ทำให้จดหมายต้นฉบับจะต้องถูกเปลี่ยนแปลงแก้ไขข้อมูลบางอย่างใน Header เช่น "From:" ซึ่งเป็น Header ที่แสดงอีเมลแอดเดรสของผู้ที่ส่งจดหมายต้นฉบับ จะต้องถูกเปลี่ยนไปเป็นอีเมลแอดเดรสของ Sublist Gateway เพื่อให้เมลลิงลิสต์ปลายทางยอมรับว่าเป็นสมาชิกสามารถส่งจดหมายได้

ทางออกของปัญหาทั้งสองคือใช้โปรโตคอล MIME (Multipurpose Internet Mail Extensions) ซึ่งขณะที่ทำการวิจัยนี้อยู่ (พ.ศ. 2538 – 2543) โปรโตคอลนี้มีสถานะเป็น Draft Standards ซึ่งเป็นสถานะสุดท้ายก่อนจะประกาศเป็นมาตรฐาน ถึงแม้ว่าโปรโตคอลนี้จะยังไม่เป็นมาตรฐานเต็มตัวแต่ก็มีความสมบูรณ์ค่อนข้างสูง ทำให้ Mail User Agent ส่วนใหญ่ในขณะนี้สนับสนุนโปรโตคอลนี้ และเชื่อแน่ว่าเมื่อโปรโตคอลนี้ได้รับการยอมรับเป็นมาตรฐานสำหรับอินเทอร์เน็ตแล้ว Mail User Agent ทุกตัวบนอินเทอร์เน็ตก็จะสนับสนุนโปรโตคอลนี้ในที่สุด

### 3.2 ความเป็นมาของโปรโตคอล MIME

RFC 822 [7] เริ่มถูกเผยแพร่มาตั้งแต่ปีคริสต์ศักราช 1982 เพื่อกำหนดรูปแบบมาตรฐานสำหรับการส่งข้อความเป็นจดหมายผ่านอินเทอร์เน็ต และประสบความสำเร็จเป็นอย่างมากดังจะเห็นได้จากการนำ RFC 822 ไปใช้อย่างแพร่หลายนอกเหนือจากอินเทอร์เน็ต เมื่อรูปแบบนี้ถูกนำไปใช้กันอย่างกว้างขวาง ข้อจำกัดหลายอย่างก็ปรากฏขึ้นในหมู่ผู้ใช้

RFC 822 ถูกเจาะจงให้กำหนดรูปแบบเฉพาะจดหมายที่เป็นข้อความ (Text) เท่านั้น ส่วนจดหมายที่ไม่ใช่ข้อความเช่น จดหมายที่มีสื่อประสม (Multimedia) เช่นภาพหรือเสียงรวมอยู่ ไม่ได้เอกลารนี้เป็นเอกลารที่สงวนไว้สำหรับการศึกษาเท่านั้น เมื่อนูญาดเห็นว่าเป็นประโยชน์ในการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกกำหนดถึง แม้ในกรณีของข้อความ RFC 822 ก็ไม่เพียงพอต่อความต้องการของผู้ใช้ที่ต้องการใช้ภาษาที่มีชุดอักขระนอกเหนือจาก US-ASCII เนื่องจาก RFC 822 ไม่ได้กำหนดกลไกสำหรับจดหมายที่บรรจุ ภาพ เสียง และข้อความที่เป็นภาษาของชาวเอเชีย

นอกจากนี้ ข้อจำกัดซึ่งเป็นที่รู้จักกันดีของระบบไปรษณีย์อิเล็กทรอนิกส์ที่ใช้โปรโตคอล RFC 821/822 คือการจำกัดข้อความของจดหมายให้ประกอบด้วยบรรทัดที่ค่อนข้างสั้น (ไม่เกิน 1000 ตัวอักษร) ของข้อความที่เป็นได้เพียง US-ASCII 7 บิต ซึ่งทำให้หากผู้ใช้ต้องการจะส่งข้อมูลที่ไม่ใช่ข้อความ ก็จำเป็นต้องแปลงข้อมูลนั้นเป็นรหัส US-ASCII 7 บิต ก่อนที่จะใช้ Mail User Agent ส่งจดหมายนั้นไป ตัวอย่างของการเข้ารหัสเพื่อให้กลายเป็น US-ASCII 7 บิต คือการส่งไปเป็นข้อความของเลขฐานสิบหก หรือแปลงโดยใช้วิธี UUEncode

โปรโตคอล MIME (Multipurpose Internet Mail Extensions) จึงเกิดขึ้น เพื่อเสนอกลไกในการแก้ปัญหาเหล่านี้โดยยังคงความเข้ากันได้กับระบบไปรษณีย์อิเล็กทรอนิกส์ที่ใช้โปรโตคอล RFC 822 โดยส่วนหลักที่ MIME กล่าวถึงคือ

1. MIME-Version header เพื่อบอกรุ่นของ MIME ที่จดหมายนั้นใช้และทำให้ Mail Agent แยกแยะความแตกต่างระหว่างจดหมายที่ใช้รูปแบบดั้งเดิม กับแบบที่ใช้ MIME
2. Content-Type header ที่นำมาจาก RFC 1049 [10] ซึ่งใช้ระบุชนิดของข้อมูลที่บรรจุอยู่ในจดหมาย
3. Content-Transfer-Encoding header ซึ่งใช้ระบุถึงวิธีการเข้ารหัสของจดหมายเพื่อใช้กับระบบไปรษณีย์อิเล็กทรอนิกส์ที่มีข้อจำกัดในเรื่องของชุดอักขระ
4. Content-ID และ Content-Description header ซึ่งใช้อธิบายลักษณะของข้อมูลในจดหมายเพิ่มเติม

Header ทุกตัวที่นิยามขึ้นมาในโปรโตคอล MIME จะเป็นไปตามกฎของ Header ที่ระบุใน RFC 822

### 3.3 MIME-Version Header

แม้ว่าโปรโตคอล MIME จะเป็นส่วนเสริมและถูกกำหนดให้มีความเข้ากันได้กับ RFC 822 แต่ก็ควรที่จะบอกให้ Mail Agent รู้ว่าจดหมายฉบับใดถูกเรียบเรียง (Compose) ขึ้นตามมาตรฐานใหม่ ดังนั้น MIME จึงนิยาม Header “MIME-Version” ขึ้นมาใหม่เพื่อบอกรุ่น (Version) ของรูปแบบที่จดหมายฉบับนั้นใช้

จดหมายที่เรียบเรียงขึ้นตาม โปรโตคอล MIME จะต้องมี Header อันหนึ่งที่เป็นไปตามนี้

MIME-Version: 1.0

ดังนั้นจดหมายที่มี Header นี้ก็แสดงว่ามีรูปแบบเป็นไปตามโปรโตคอล MIME ซึ่งในขณะที่ทำการวิจัยนี้ MIME ก็ยังคงไม่เปลี่ยนเลขรุ่น แต่ก็อาจจะเพิ่มรุ่นขึ้นได้ในอนาคตหากมีการกำหนด MIME รูปแบบใหม่ออกมา นอกจากนี้ยังต้องมี Header นี้ในส่วน Top level ของจดหมาย แต่ไม่จำเป็นต้องมีในส่วน body แต่ละส่วนของ Multipart entity และ Header นี้จำเป็นต้องมีในส่วน Header ที่ฝังตัวอยู่ใน body ที่เป็นประเภท “message/rfc822” ถ้าจดหมายที่ฝังตัวอยู่นั้นเป็นไปตามโปรโตคอล MIME

### 3.4 Content-ID Header

ส่วน Body ของจดหมายแบบ MIME อาจจะมีการอ้างอิงถึงส่วนอื่นๆ ได้ ดังนั้นส่วน Body ควรจะถูกอ้างอิงได้โดยใช้ “Content-ID” header ซึ่งจะมีรูปแบบเช่นเดียวกับ “Message-Id” header ตาม RFC 822 และเช่นเดียวกับ Message-Id แต่ละ Content-Id ต้องถูกสร้างขึ้นให้มีหนึ่งเดียวในโลก (World-unique) ไม่ให้ซ้ำกับใคร

Content-ID header มักใช้ในการแคช (Caching) ข้อมูลที่ถูกอ้างอิงโดย Message/external-body content-type เนื่องจากเป็น Header ที่ต้องมีสำหรับแต่ละ Message/external-body entity

### 3.5 Content-Type Header

จุดประสงค์ของ Content-Type header ก็บอกลักษณะของข้อมูลที่บรรจุอยู่ใน Body ของจดหมาย เพื่อให้ Mail User Agent สามารถเลือกวิธีที่จะแสดงข้อมูลเหล่านั้นต่อผู้ใช้ได้ สิ่งนี้ที่ Header นี้แสดงเรียกว่า Media type

Content-Type Header จะระบุชนิดของข้อมูลในส่วน Body โดยแสดงเป็น Media Type และ Subtype ตามด้วยชุดของพารามิเตอร์ในรูปแบบ attribute=value โดยไม่จำเป็นต้องเรียงตามลำดับ โดยมีรูปแบบดังนี้

```
Content-Type: type/subtype; attribute1=value1;
attribute2=value2;...
```

โดยที่ Type จะระบุถึงชนิดของข้อมูลอย่างกว้างๆ ขณะที่ Subtype จะระบุรูปแบบเฉพาะของข้อมูลชนิดนั้น โดยทั้ง Type และ Subtype นั้นจำเป็นต้องระบุเสมอสำหรับ Content-Type Header ส่วนชุดของพารามิเตอร์นั้นจะแตกต่างกัน และอาจไม่จำเป็นต้องมี ขึ้นอยู่กับ Media Type และ Subtype ตัวอย่างเช่น จดหมายที่เป็นไปตาม RFC 822 ดังเดิมซึ่งเป็นข้อความธรรมดาและใช้ชุดอักขระ US-ASCII นั้นสามารถแสดงได้ด้วย Content-Type ดังนี้

```
Content-type: text/plain; charset=us-ascii
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานส่งเสริมการค้าในต่างประเทศ ณ นครเชียงใหม่ โดยสงวนลิขสิทธิ์ไว้เพื่อใช้ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งแสดงว่า Body ของจดหมายฉบับนี้มี Media type ชนิด Text และมี Subtype เป็น Plain โดยมีพารามิเตอร์ Charset เป็น Us-ascii ซึ่ง Content-Type ดังตัวอย่างนี้ยังถือเป็นค่าโดยปริยาย (Default) หากไม่ปรากฏ Content-Type Header ในจดหมายแบบ MIME

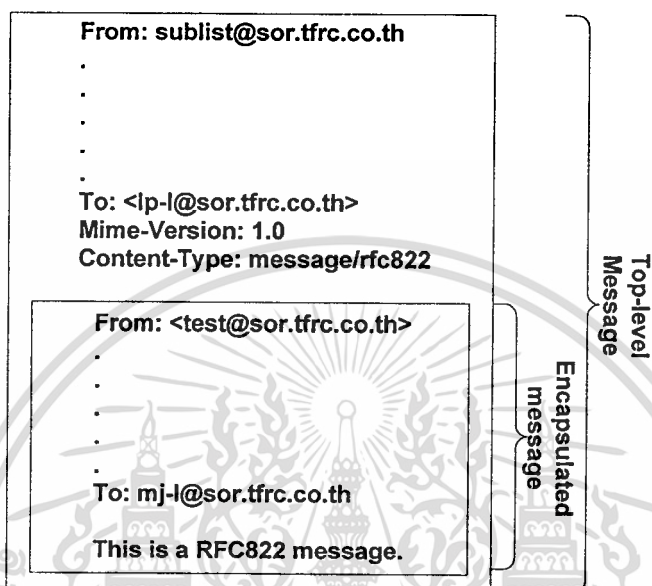
### 3.6 Media Types

Media Type แบ่งเป็นสองประเภทหลักได้แก่

1. **Discrete Media Type** คือ Media type ที่ต้องถูกประมวลผลแยกเป็นอิสระจากโปรโตคอล MIME นั่นคืออาจต้องใช้กลไกอื่นนอกเหนือจาก MIME มาช่วยในการแสดงผล Media type ที่จัดอยู่ในประเภทนี้คือ
  - 1.1. **Text** คือข้อมูลที่เป็นข้อความ Subtype “plain” จะแสดงว่าเป็นข้อความธรรมดาไม่มีคำสั่งในการกำหนดรูปแบบอื่นใดของข้อความ สามารถแสดงผลได้ทันทีไม่ต้องใช้ซอฟต์แวร์ใดๆ มาช่วย Subtype อื่นๆ ได้แก่ “enriched” (RFC 1896) ซึ่งใช้กับข้อความที่มีการจัดรูปแบบให้สวยงามยิ่งขึ้น ทำให้อาจต้องใช้ซอฟต์แวร์อื่นมาช่วยในการแสดงผล
  - 1.2. **Image** คือข้อมูลที่เป็นรูปภาพ ซึ่งต้องการอุปกรณ์ในการแสดงผลที่สามารถแสดงผลแบบกราฟิกในการดูข้อมูล Subtype ที่ใช้กันอย่างกว้างขวางคือ JPEG
  - 1.3. **Audio** คือข้อมูลเสียง ซึ่งต้องการอุปกรณ์ที่สามารถส่งเสียงได้ เช่น ลำโพงหรือโทรศัพท์
  - 1.4. **Video** คือข้อมูลที่เป็นภาพเคลื่อนไหว ซึ่งต้องการอุปกรณ์ในการแสดงผลที่มักรวมทั้งฮาร์ดแวร์และซอฟต์แวร์ ตัวอย่างเช่น Subtype “mpeg”
  - 1.5. **Application** คือข้อมูลชนิดอื่นๆ ที่ต้องถูกประมวลผลโดยโปรแกรมอื่น เช่น Subtype “octet-stream” และ “PostScript”
2. **Composite Media Type** คือ Media type ที่ต้องถูกประมวลผลต่อตามกลไกของ MIME โดยตรง Media type ที่จัดอยู่ในประเภทนี้คือ
  - 2.1. **Multipart** คือข้อมูลที่ประกอบด้วย Entity ของข้อมูลหลายๆ ชนิดแยกส่วนเป็นอิสระกัน มีสี่ Subtype ในเบื้องต้นคือ “mixed” สำหรับแต่ละส่วนที่ไม่เกี่ยวข้องกัน “alternative” ที่แต่ละส่วนเป็นข้อมูลเดียวกันแต่ต่างรูปแบบ “parallel” ที่สามารถดูทุกส่วนไปพร้อมๆ กัน และ “digest” ที่แต่ละส่วนเป็นข้อมูลชนิด “message/rfc822”
  - 2.2. **Message** คือข้อมูลที่เป็นจดหมายที่ถูกห่อหุ้มเอาไว้ (Encapsulated) ซึ่งอาจจะเป็นบางส่วนหรือทั้งหมดของจดหมาย Message media type เป็น Media type ที่จะนำมาประยุกต์ใช้ในระบบจัดการเมลลิ่งลิสต์ย่อย เนื่องจากมันจะรักษาข้อมูลของจดหมายต้นฉบับไว้ ทำให้ Sublist gateway สามารถเปลี่ยนแปลงแก้ไข Top-level header ได้โดยไม่เสียข้อมูลของ Header เดิมไป

### 3.6.1 RFC822 Subtype

จดหมายที่มี Media type เป็นแบบ “message/rfc822” แสดงว่าส่วน Body บรรจุจดหมายอิเล็กทรอนิกส์ไว้ โดยจดหมายที่ถูกบรรจุไว้ต้องมี Header “From” “Subject” หรือ “Date” อย่างน้อยหนึ่ง Header [11] ดังรูปที่ 3.1



รูปที่ 3.1 รูปแบบของจดหมายแบบ Message/RFC822

ถึงแม้ว่าจะหมายเลข “822” จดหมายที่ถูกห่อหุ้มอยู่ก็ไม่ได้จำกัดว่าจะต้องมีรูปแบบเฉพาะ RFC 822 แต่สามารถเป็นบทความข่าว (News article) หรือเป็นจดหมายแบบ MIME ก็ได้

Sublist gateway ของระบบจัดการเมลลิงลิสต์ย่อยจะทำการแปลงจดหมายจากเมลลิงลิสต์ย่อยโดยใช้รูปแบบ Message/RFC822 นี้ก่อนส่งต่อให้เมลลิงลิสต์หลัก เพื่อรักษาด้านฉบับเดิมของจดหมายจากเมลลิงลิสต์ย่อยเอาไว้ไม่ให้ถูกกระทบจากการแก้ไข Header เพื่อให้เมลลิงลิสต์หลักยอมรับว่าเป็นจดหมายของสมาชิก โดยรูปที่ 3.2 ต่อไปนี้เป็นตัวอย่างจดหมายต้นฉบับจากเมลลิงลิสต์ย่อย ก่อนที่จะถูกแปลงเพื่อส่งต่อให้เมลลิงลิสต์หลักด้วย Sublist gateway

```

From owner-major-l@sor.tfrc.co.th  Fri Mar 24 02:24:39 2000
Return-Path: <owner-major-l@sor.tfrc.co.th>
Received: (from majordom@localhost)
    by sor.tfrc.co.th (8.8.5/8.8.5) id CAA00268
    for major-l-list; Fri, 24 Mar 2000 02:24:39 +0700
X-Authentication-Warning: sor.tfrc.co.th: majordom set sender to
    owner-major-l@sor.tfrc.co.th using -f
Received: from localhost (user@localhost)
    by sor.tfrc.co.th (8.8.5/8.8.5) with SMTP id CAA00262;
    Fri, 24 Mar 2000 02:24:35 +0700
Date: Fri, 24 Mar 2000 02:24:35 +0700 (ICT)
From: General User <user@sor.tfrc.co.th>
To: For Testing <test@sor.tfrc.co.th>
cc: major-l@sor.tfrc.co.th
Subject: Re: From remote mj-l
In-Reply-To: <Pine.LNX.3.96.1000324022113.223A-100000@sor.tfrc.co.th>
Message-ID: <Pine.LNX.3.96.1000324022358.258A-100000@sor.tfrc.co.th>
MIME-Version: 1.0
Content-Type: TEXT/PLAIN; charset=US-ASCII
Sender: owner-major-l@sor.tfrc.co.th
Precedence: bulk

This is the body of this message...

```

**รูปที่ 3.2** ตัวอย่างจดหมายต้นฉบับจากเมลลิงลิสต์ย่อย ก่อนที่จะถูกแปลงด้วย Sublist gateway

เมื่อจดหมายดังกล่าวข้างต้นผ่านการแปลงโดย Sublist gateway แล้ว จะได้ผลลัพธ์ดัง  
ตัวอย่างในรูปที่ 3.3 ต่อไปนี้

```

1. From owner-mj-l@sor.tfrc.co.th Fri Mar 24 02:24:48 2000
2. Return-Path: <owner-mj-l@sor.tfrc.co.th>
3. Received: (from majordom@localhost)
4.   by sor.tfrc.co.th (8.8.5/8.8.5) id CAA00284
5.   for mj-l-list; Fri, 24 Mar 2000 02:24:48 +0700
6. Received: from sor.tfrc.co.th (sublist@localhost [127.0.0.1])
7.   by sor.tfrc.co.th (8.8.5/8.8.5) with SMTP id CAA00279
8.   for <mj-l@sor.tfrc.co.th>; Fri, 24 Mar 2000 02:24:46 +0700
9. X-Authentication-Warning: sor.tfrc.co.th: majordom set sender to
10.  owner-major-l@sor.tfrc.co.th using -f
11. Date: Fri, 24 Mar 2000 02:24:35 +0700 (ICT)
12. From: sublist@sor.tfrc.co.th
13. Subject: Re: From remote mj-l
14. In-Reply-To: <Pine.LNX.3.96.1000324022113.223A-100000@sor.tfrc.co.th>
15. Message-Id: <Pine.LNX.3.96.1000324022358.258A-100000@sor.tfrc.co.th>
16. Mime-Version: 1.0
17. Content-Type: message/rfc822
18. To: mj-l@sor.tfrc.co.th
19. Reply-To: General User <user@sor.tfrc.co.th>
20. X-Mailer: Perl5 Mail::Internet v1.30
21. Sender: owner-mj-l@sor.tfrc.co.th
22. Precedence: bulk

23. >From owner-major-l@sor.tfrc.co.th Fri Mar 24 02:24:39 2000
24. Received: (from majordom@localhost) by sor.tfrc.co.th (8.8.5/8.8.5) id
25.   CAA00268 for major-l-list; Fri, 24 Mar 2000 02:24:39 +0700
26. X-Authentication-Warning: sor.tfrc.co.th: majordom set sender to
27.   owner-major-l@sor.tfrc.co.th using -f
28. Received: from localhost (user@localhost) by sor.tfrc.co.th
29.   (8.8.5/8.8.5) with SMTP id CAA00262; Fri, 24 Mar 2000 02:24:35
30.   +0700
31. Date: Fri, 24 Mar 2000 02:24:35 +0700 (ICT)
32. From: General User <user@sor.tfrc.co.th>
33. To: For Testing <test@sor.tfrc.co.th>
34. Cc: major-l@sor.tfrc.co.th
35. Subject: Re: From remote mj-l
36. In-Reply-To: <Pine.LNX.3.96.1000324022113.223A-100000@sor.tfrc.co.th>
37. Message-Id: <Pine.LNX.3.96.1000324022358.258A-100000@sor.tfrc.co.th>
38. Mime-Version: 1.0
39. Content-Type: TEXT/PLAIN; charset=US-ASCII
40. Sender: owner-major-l@sor.tfrc.co.th
41. Precedence: bulk

42. This is the body of this message...

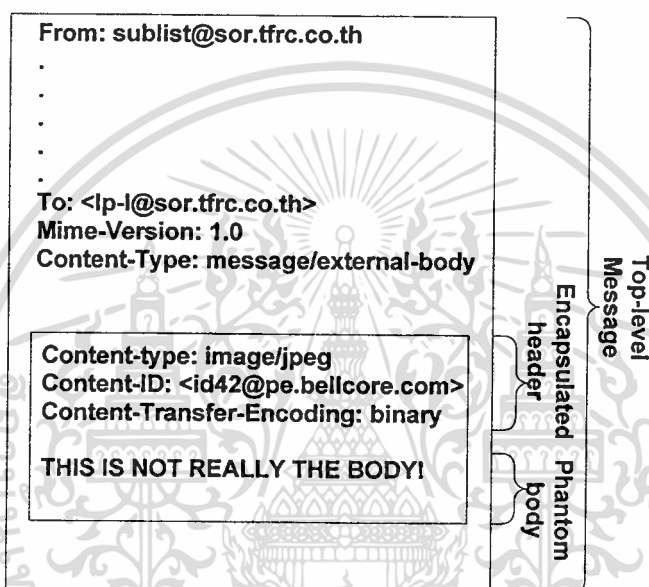
```

### รูปที่ 3.3 ตัวอย่างจดหมายที่ถูกแปลงด้วย Sublist gateway โดยใช้รูปแบบ Message/RFC822

จากตัวอย่างแสดงให้เห็นว่า Sublist gateway จะทำการเปลี่ยนแปลง Header “From:” ในบรรทัดที่ 12 เป็นอีเมลแอดเดรสของ Sublist gateway โดยใส่อีเมลแอดเดรสของผู้ส่งจดหมายต้นฉบับไว้ที่ Header “Reply-To” ในบรรทัดที่ 19 เพื่อให้สมาชิกในเมลลิ่งลิสต์หลักสามารถตอบจดหมายถึงผู้ส่งที่แท้จริงได้โดยตรงถ้าต้องการ และเปลี่ยนเป็นจำหน้าถึงเมลลิ่งลิสต์หลักแทนในบรรทัดที่ 18 จากนั้นจึงแปลงจดหมายต้นฉบับไปเป็นรูปแบบ message/rfc822 ดังบรรทัดที่ 17 ซึ่งจดหมายต้นฉบับจะถูกห่อหุ้มอยู่ในตั้งแต่บรรทัดที่ 23 ถึง 42

### 3.6.2 External-Body Subtype

External-body subtype แสดงว่าข้อมูลของ Body ที่แท้จริงไม่ได้รวมอยู่ในจดหมาย แต่มีเพียงส่วนที่อ้างถึงเท่านั้น ซึ่งจะมีพารามิเตอร์ที่อธิบายวิธีการเข้าถึง Body นั้น จดหมายแบบ “message/external-body” จะประกอบด้วยส่วน Header ตามด้วยหนึ่งบรรทัดว่าง ตามด้วย Header ของจดหมายที่ถูกห่อหุ้มอยู่ นอกจากนี้ยังอาจตามด้วยอีกหนึ่งบรรทัดว่างและส่วน Body หลอกๆ เนื่องจาก Body จริงจะอยู่ที่อื่น Body หลอกๆจะมีประโยชน์เฉพาะบาง access-type เช่น “mail-server” ดังรูปที่ 3.4



รูปที่ 3.4 รูปแบบของจดหมายแบบ Message/External-body

Header ของจดหมายที่ถูกห่อหุ้มอยู่ต้องมี Content-ID header เพื่อใช้อ้างถึงข้อมูล ซึ่งอาจใช้ในการทำแคช หรือเพื่อใช้จดจำข้อมูลที่รับมาเมื่อ access-type เป็น “mail-server”

พารามิเตอร์ทั่วไปของ External-Body คือ

1. **ACCESS-TYPE** แสดงวิธีการเข้าถึงข้อมูล เช่น “FTP” “ANON-FTP” “URL” “LOCAL-FILE” และ “MAIL-SERVER” พารามิเตอร์นี้เป็นพารามิเตอร์บังคับ
2. **EXPIRATION** บอกวันที่ที่หลังจากนี้ไม่รับรองว่า Body ที่อ้างถึงจะยังคงอยู่ อาจใช้กับ access-type ใดก็ได้ และไม่ใช้พารามิเตอร์บังคับ
3. **SIZE** บอกขนาดของข้อมูลเพื่อจะได้เตรียมเนื้อที่ไว้รองรับเมื่อดึงข้อมูลมา อาจใช้กับ access-type ใดก็ได้ และไม่ใช้พารามิเตอร์บังคับ
4. **PERMISSION** แสดงว่าข้อมูลที่ดึงมาสามารถถูกเขียนทับได้หรือไม่ อาจใช้กับ access-type ใดก็ได้ และไม่ใช้พารามิเตอร์บังคับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Access-type ที่ระบบจัดการเมลลิงลิสต์ย่อยใช้คือ "URL" [12] ซึ่งมีพารามิเตอร์ บังคับคือ "URL" เช่นกัน โดยมีรูปแบบดังนี้

```
Content-type: message/external-body; access-type=URL;
URL="URL string"
```

โดยรูปแบบของ URL string ตาม RFC 1738 [13] นั้นไม่จำกัดความยาวและอักขระซึ่ง อาจเป็นปัญหาเกี่ยวกับข้อกำหนดของการตัดขึ้นบรรทัดใหม่ (Wrapped) ของ Header ตาม RFC 822 ดังนั้นจึงต้องแก้ไขตามขั้นตอนต่อไปนี้

1. เข้ารหัส SPACE CTL double quotes backslash และ อักขระ 8 บิต ตามการเข้ารหัสของ URL ที่ระบุใน RFC 1738
2. แยก URL ที่ได้ออกเป็น String ย่อยขนาดไม่เกิน 40 ตัวอักษร
3. คั่น String ย่อยที่ได้ด้วยช่องว่างอย่างน้อยหนึ่งช่อง

ระบบจัดการเมลลิงลิสต์ย่อยใช้รูปแบบ message/external-body และ access-type=URL กับจดหมายที่ส่งให้สมาชิกของเมลลิงลิสต์ย่อย เพื่อลดความซ้ำซ้อนในการเก็บจดหมายของสมาชิกในเมลลิงลิสต์ย่อย โดยเก็บจดหมายต้นฉบับจากเมลลิงลิสต์หลักไว้ที่แม่ข่ายส่วนกลาง และให้สมาชิกของเมลลิงลิสต์ย่อยดึงไปอ่านได้ตาม URL ที่ส่งให้ ดังตัวอย่างในรูปที่ 3.5 ต่อไปนี้

```
1. From owner-major-1@sor.tfrc.co.th Fri Mar 24 02:22:01 2000
2. Return-Path: <owner-major-1@sor.tfrc.co.th>
3. Received: (from majordom@localhost)
4.   by sor.tfrc.co.th (8.8.5/8.8.5) id CAA00245
5.   for major-1-list; Fri, 24 Mar 2000 02:22:01 +0700
6. Received: from sor.tfrc.co.th (sublist@localhost [127.0.0.1])
7.   by sor.tfrc.co.th (8.8.5/8.8.5) with SMTP id CAA00240
8.   for <major-1@sor.tfrc.co.th>; Fri, 24 Mar 2000 02:21:59 +0700
9. X-Authentication-Warning: sor.tfrc.co.th: majordom set sender to
10.  owner-mj-1@sor.tfrc.co.th using -f
11. Date: Fri, 24 Mar 2000 02:21:49 +0700 (ICT)
12. From: sublist@sor.tfrc.co.th
13. Subject: From remote mj-1
14. Message-Id: <Pine.LNX.3.96.1000324022113.223A-100000@sor.tfrc.co.th>
15. Mime-Version: 1.0
16. Content-Type: message/external-body; access-type=URL;
17.   URL="http://sor.tfrc.co.th/~sublist/mj-1%40so
18.   r.tfrc.co.th/Pine.LNX.3.96.1000324022113
19.   .223A-100000%40sor.tfrc.co.th"
20. To: major-1@sor.tfrc.co.th
21. Reply-To: For Testing <test@sor.tfrc.co.th>
22. X-Mailer: Perl5 Mail::Internet v1.30
23. Sender: owner-major-1@sor.tfrc.co.th
24. Precedence: bulk

25. Content-Type: message/rfc822
26. Content-Id: <SL.Pine.LNX.3.96.1000324022113.223A-100000@sor.tfrc.co.th>
```

รูปที่ 3.5 ตัวอย่างจดหมายที่ถูกแปลงด้วย Sublist gateway เป็นรูปแบบ Message/External-Body

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในบรรทัดที่ 16 ถึง 19 แสดงการใช้รูปแบบ message/external-body และ access-type=URL ส่วนบรรทัดที่ 25 ถึง 26 เป็น Header ของจดหมายที่ถูกห่อหุ้มอยู่ซึ่งเป็นแบบ message/rfc822 แต่ไม่มีส่วน body ที่แท้จริงซึ่งสามารถดึงมาอ่านได้จาก URL “http://sor.tfrc.co.th/~sublist/mj-l@sor.tfrc.co.th/Pine.LNX.3.96.1000324022113.223A-100000@sor.tfrc.co.th” ซึ่งระบุในบรรทัดที่ 16 ถึง 19

### 3.7 สรุป

เมื่อ Sublist Gateway ได้รับจดหมายจากเมลลิงลิสต์หลัก ก็จะเก็บจดหมายต้นฉบับนั้นไว้ที่แม่ข่ายส่วนกลาง แล้วส่งจดหมายที่มี Content-Type แบบ Message/External-Body ให้กับเมลลิงลิสต์ย่อยเพื่อใช้อ้างอิงกลับไปหาจดหมายต้นฉบับนั้น

จะเห็นได้ว่าการจัดเก็บและส่งจดหมายตามโพรโทคอล MIME นี้จะช่วยลดเนื้อที่ที่ใช้เก็บจดหมายที่ซ้ำซ้อนกันลงไปได้ เพราะ Body ของเมลจริงๆ จะเก็บอยู่ที่แม่ข่ายส่วนกลางเพียงสำเนาเดียว สำหรับจดหมายที่ส่งให้ผู้รับแต่ละคนจะเป็นเพียงส่วนที่อ้างอิงกลับไปหา Body จริงๆ ที่ส่วนกลางนี้เท่านั้น ซึ่งจะมีเพียง Header ขนาดเพียงไม่กี่ไบต์ซึ่งจะเป็นการประหยัดเนื้อที่ในการเก็บจดหมายลงไปได้มากเมื่อเทียบกับการที่ต้องส่ง Body จริงๆ ที่อาจมีขนาดหลายพันไบต์ไปให้ผู้รับทุกๆ คน

นอกจากนี้โพรโทคอล MIME ยังช่วยรักษา Header ของจดหมายต้นฉบับไว้ไม่ให้สูญหายไปกับการแก้ไข Header ของ Sublist Gateway เพื่อให้สามารถส่งต่อจดหมายให้กับเมลลิงลิสต์หลักได้ โดยใช้ Content-Type แบบ Message/RFC822

## บทที่ 4

### การออกแบบและการทำงานของระบบจัดการเมลถึงลิสต์ย่อย

จากหลักการทั้งหมดที่ได้กล่าวมาแล้วในบทก่อนๆ สามารถออกแบบระบบจัดการเมลถึงลิสต์ย่อยโดยมีขั้นตอนต่างดังต่อไปนี้

#### 4.1 Requirement Specification

จากสมมุติฐานของการศึกษาที่ใช้แก้ปัญหาต่างๆ ของเมลถึงลิสต์สามารถกำหนด Requirement Specification ได้ดังนี้

1. สามารถถ่ายทอดจดหมายระหว่างเมลถึงลิสต์หลักและเมลถึงลิสต์ย่อยใดๆ ที่จัดการด้วยซอฟต์แวร์จัดการเมลถึงลิสต์ต่างๆ ไปได้ทั้งสองทิศทาง โดยไม่ต้องแก้ไขซอฟต์แวร์จัดการเมลถึงลิสต์เหล่านั้น
2. สามารถรักษาข้อมูลของจดหมายต้นฉบับ เช่น Header ของผู้ส่งที่แท้จริง หากต้องแก้ไขจดหมายเพื่อให้สามารถส่งข้ามไปยังอีกเมลถึงลิสต์หนึ่งได้
3. สามารถนำจดหมายจากเมลถึงลิสต์หลักมาส่งให้สมาชิกในเมลถึงลิสต์ย่อยได้ตามโปรโตคอล MIME โดยใช้ Message Media Type และ External-Body Subtype (Content-type: message/external-body)
4. สามารถลบจดหมายต้นฉบับที่ส่วนกลางตามโปรโตคอล MIME ที่ใช้ Content-type: message/external-body ได้ตามระยะเวลาที่กำหนด
5. สามารถป้องกันการเกิด Mailing loop ได้

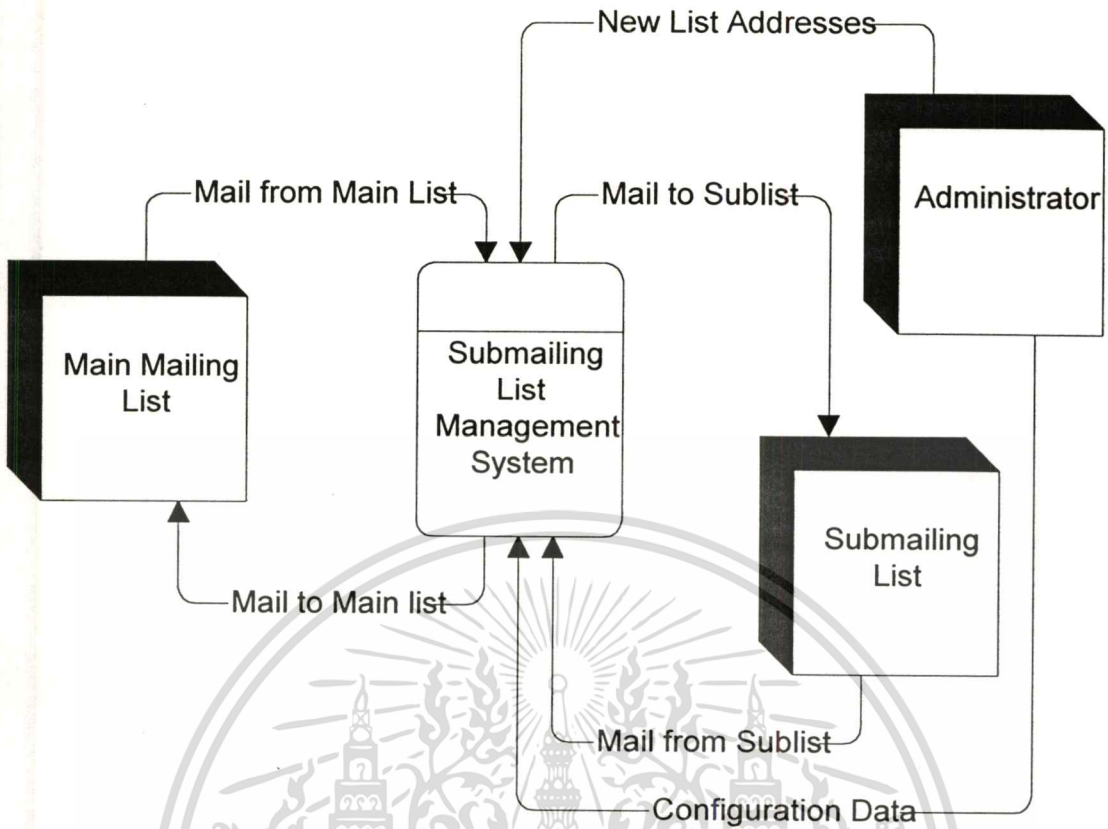
#### 4.2 Systems Analysis

##### 4.2.1 Data Flow Analysis

จากความต้องการของระบบ สามารถวิเคราะห์กระแสข้อมูลซึ่งแสดงได้ด้วย Data flow diagram ได้ดังนี้

##### 4.2.1.1 Context Diagram

Context diagram ของระบบจัดการเมลถึงลิสต์ย่อยซึ่งแสดงส่วนหลักของต้นทางและปลายทางของข้อมูลแสดงได้ดังรูปที่ 4.1



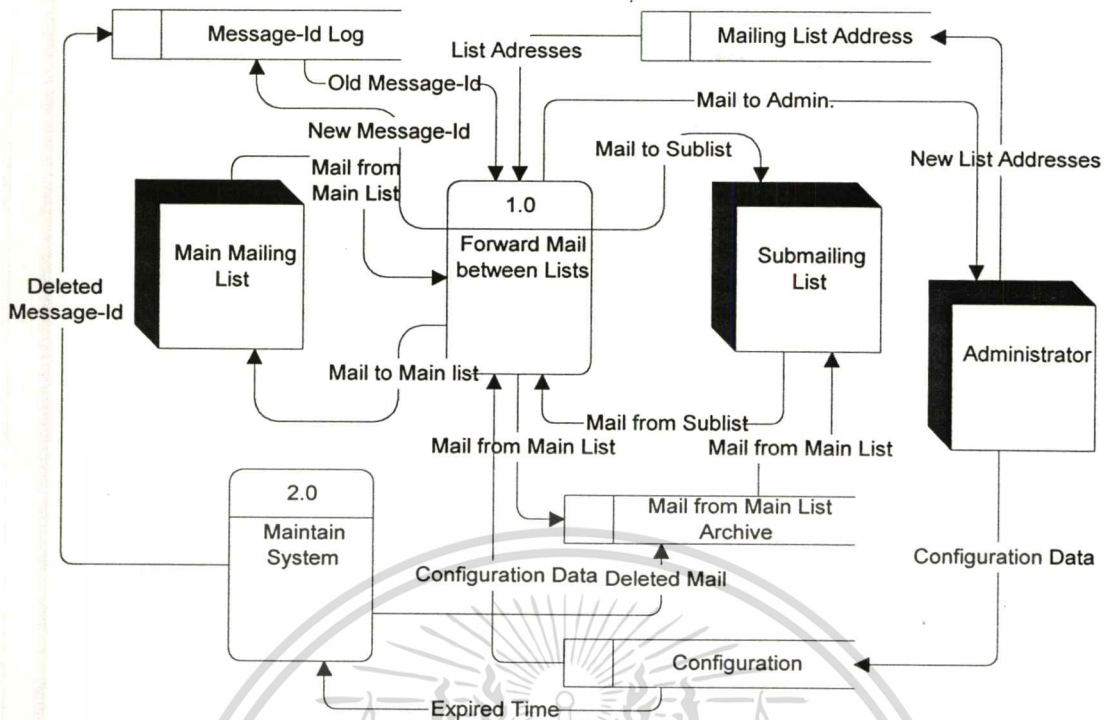
รูปที่ 4.1 Context diagram ของระบบจัดการเมลลิงลิสต์ย่อย

#### 4.2.1.2 Level-0 Data Flow Diagram

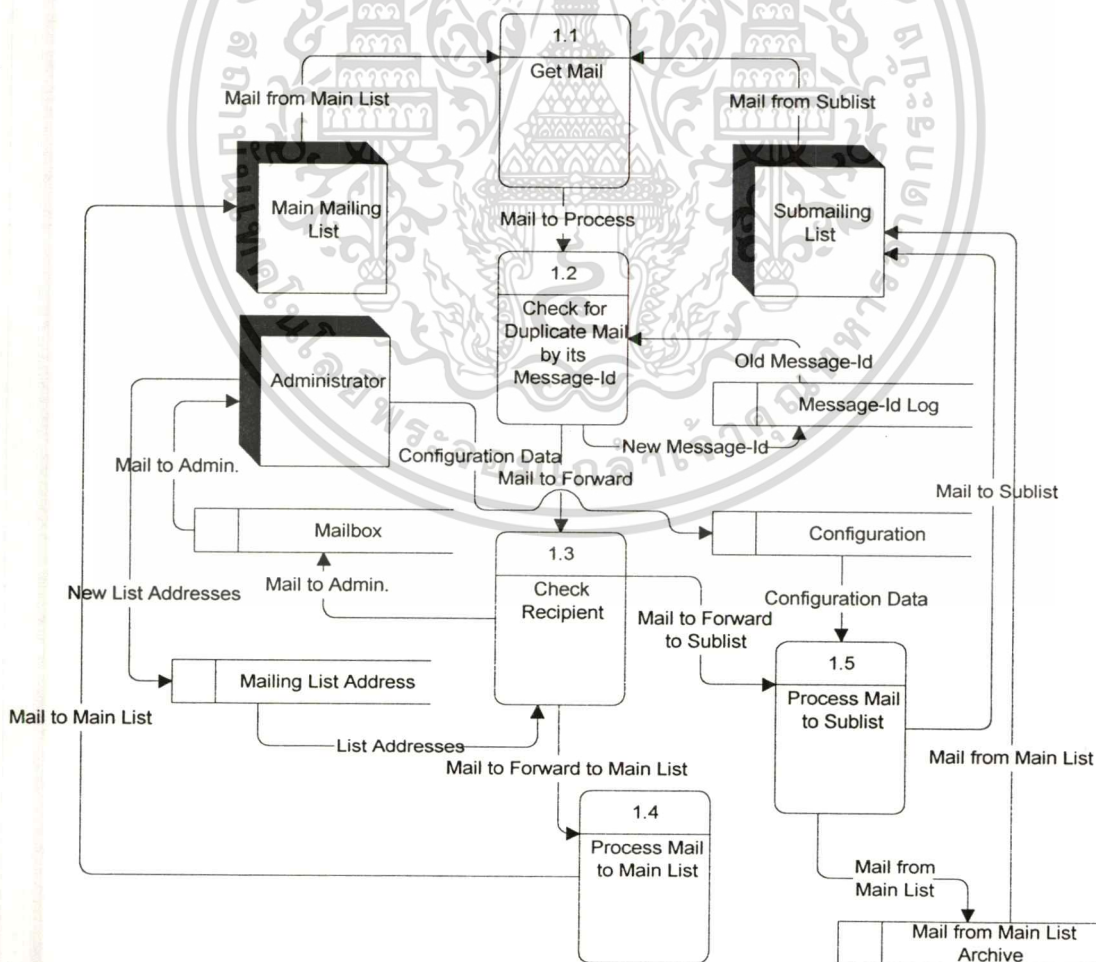
จาก Context diagram ที่ได้แสดงมา Level-0 data flow diagram ของระบบจัดการเมลลิงลิสต์ย่อยจะแสดงหน้าที่หลักๆ ของระบบดังรูปที่ 4.2

#### 4.2.1.3 Level-1 Data Flow Diagram

จาก Level-0 data flow diagram ของระบบจัดการเมลลิงลิสต์ย่อย ใน transform ที่ 1.0 (Forward Mail between Lists) ยังแสดงข้อมูลไม่เพียงพอต่อการออกแบบ ดังนั้นจึงต้องจำแนกออกเป็น Level-1 Data Flow Diagram ซึ่งแสดงรายละเอียดของขั้นตอนในการส่งต่อจดหมายระหว่างเมลลิงลิสต์ได้ดังรูปที่ 4.3



รูปที่ 4.2 Level-0 Data Flow Diagram ของระบบจัดการเมลลิงลิสต์ย่อย



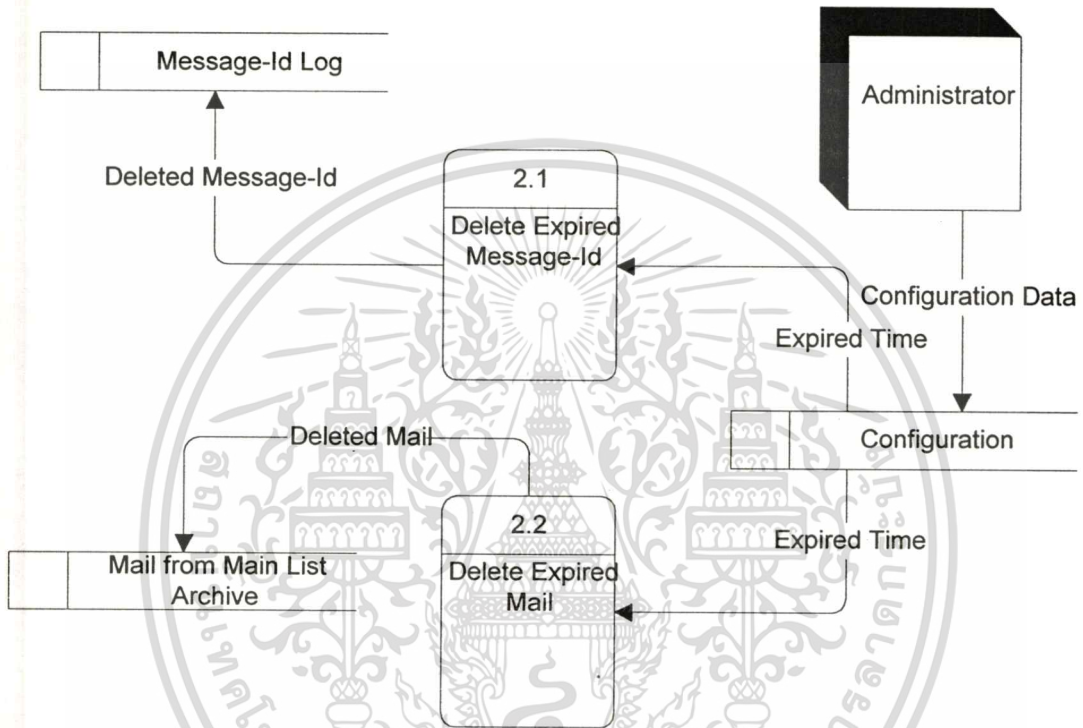
รูปที่ 4.3 Level-1 Data Flow Diagram ของขั้นตอนส่งต่อจดหมายระหว่างเมลลิงลิสต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.2.1.4 Level-2 Data Flow Diagram**

และเช่นเดียวกัน จาก Level-0 data flow diagram ของระบบจัดการเมลถึงลิสต์ย่อย ใน transform ที่ 2.0 (Maintain System) ยังแสดงข้อมูลไม่เพียงพอต่อการออกแบบ ดังนั้นจึงต้องจำแนกออกเป็น Level-2 Data Flow Diagram ซึ่งแสดงรายละเอียดของขั้นตอนในการบำรุงรักษาระบบจัดการเมลถึงลิสต์ย่อยได้ดังรูปที่ 4.4



รูปที่ 4.4 Level-2 Data Flow Diagram ของขั้นตอนบำรุงรักษาระบบ

**4.2.2 Data Dictionary**

Data dictionary เป็นการอธิบายนิยามและความหมายของข้อมูลต่างๆ ที่แสดงใน Data flow diagram อันได้แก่นิยามของกระแสข้อมูล (Data flow definition) และโครงสร้างข้อมูลที่เก็บใน Data store ต่างๆ โดยมีรายละเอียดซึ่งแสดงด้วย DeMacro data dictionary conventions [14] ดังต่อไปนี้

**4.2.2.1 Data Flow Definition**

Data flow definition เป็นการอธิบายนิยามของกระแสข้อมูลต่างๆ ที่แสดงใน Data flow diagram โดยมีรายละเอียดดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Deleted Mail = ข้อมูลในการลบจดหมายต้นฉบับจากเมลลิงลิสต์หลักที่เก็บไว้ที่ส่วนกลาง

Deleted Message-Id = ข้อมูลในการลบ Message-Id

Expired Time = ระยะเวลาที่จะเก็บรักษาจดหมายต้นฉบับจากเมลลิงลิสต์หลักที่ส่วนกลาง

List Addresses = [อีเมลแอดเดรสของเมลลิงลิสต์หลัก/อีเมลแอดเดรสของเมลลิงลิสต์ย่อย]

Mail from Main List = จดหมายต้นฉบับที่ส่งจากเมลลิงลิสต์หลัก

Mail from Sublist = จดหมายจากเมลลิงลิสต์ย่อย

Mail to Admin. = จดหมายที่ไม่ได้ส่งถึงเมลลิงลิสต์ที่จะต้องทำการส่งต่อ

Mail to Forward = จดหมายที่จะต้องทำการส่งต่อระหว่างเมลลิงลิสต์

Mail to Forward to Main List = จดหมายที่จะต้องทำการส่งต่อให้เมลลิงลิสต์หลัก

Mail to Forward to Sublist = จดหมายที่จะต้องทำการส่งต่อให้เมลลิงลิสต์ย่อย

Mail to Main List = จดหมายจากเมลลิงลิสต์ย่อยที่ผ่านการแก้ไขเพื่อส่งให้เมลลิงลิสต์หลักแล้ว

Mail to Process = จดหมายที่กำลังจะประมวลผล

Mail to Sublist = จดหมายจากเมลลิงลิสต์หลักที่ผ่านการแก้ไขเพื่อส่งให้เมลลิงลิสต์ย่อยแล้ว

New List Addresses = อีเมลแอดเดรสของเมลลิงลิสต์หลัก + อีเมลแอดเดรสของเมลลิงลิสต์ย่อย  
ของเมลลิงลิสต์หลักนั้นที่ผู้ดูแลระบบต้องกำหนดเมื่อสร้างเมลลิงลิสต์ย่อย  
ลิสต์ใหม่

New Message-Id = Message-Id ของจดหมายที่กำลังจะได้รับการประมวลผลแล้ว

Old Message-Id = Message-Id ของจดหมายที่เคยได้รับการประมวลผลแล้ว

#### 4.2.2.2 Data Store Descriptions

นอกจาก Data flow definition แล้ว ต่อไปนี้จะเป็นการอธิบายโครงสร้างของข้อมูลที่เก็บ  
อยู่ใน Data store หรือเพิ่มข้อมูลต่างๆ

Configuration = 1 {Configuration Parameter}n

Configuration Parameter = Parameter + Value

Mailbox = 1 {Mail to Admin.}n

Mail from Main List Archive = 1 {Mail from Main List Directory}n

Mail from Main List Directory = 1 {Mail from Main List File}n

Mail from Main List File = เพิ่มข้อมูลที่เก็บจดหมายจากเมลลิงลิสต์หลักแต่ละฉบับ

Mailing List Address = 1 {Mailing List Address Record}n

Mailing List Address Record = Main Mailing List Address + Submailing List Address

Message-Id Log = 1 {Message-Id Log Record}n

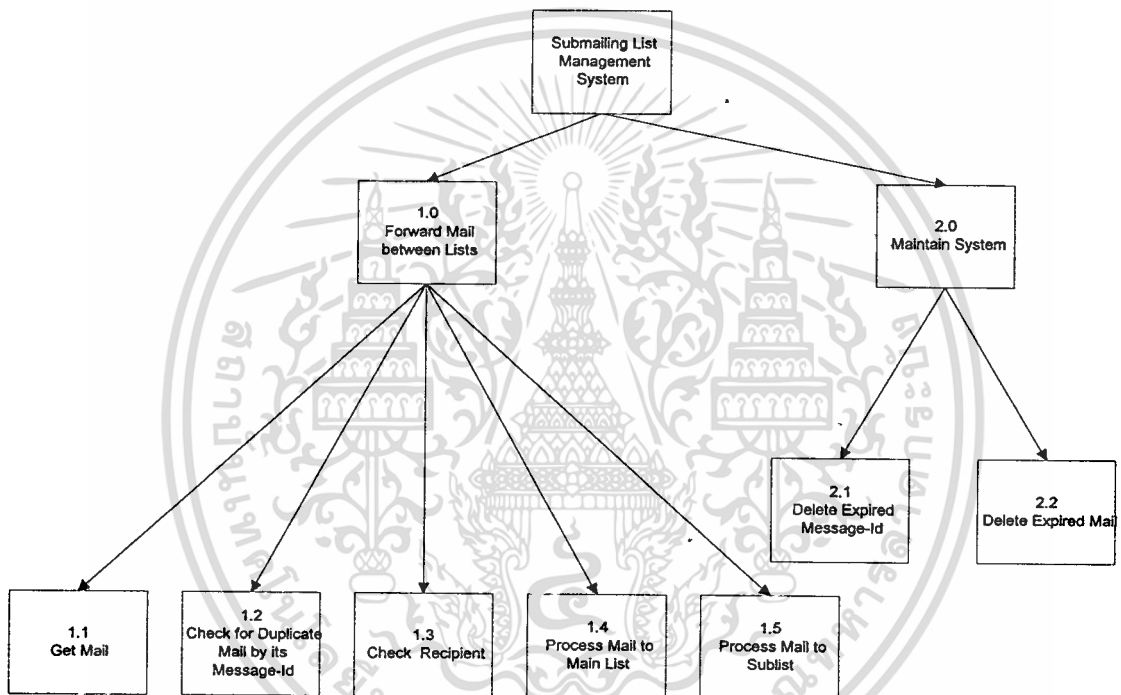
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Message-Id Log Record = Message-Id + Timestamp

### 4.3 การออกแบบระบบ

#### 4.3.1 โครงสร้างของระบบ

จากการวิเคราะห์ระบบโดยใช้ Data flow diagram สามารถออกแบบโครงสร้างของระบบจัดการเมลลิงลิสต์ย่อยซึ่งแสดงเป็น Structure chart ดังรูปที่ 4.5 ซึ่งแสดงให้เห็นว่า ระบบจัดการเมลลิงลิสต์ย่อยแบ่งออกเป็นสองส่วนหลักๆ ซึ่งทำหน้าที่แยกกันดังนี้



รูปที่ 4.5 Structure Chart ของระบบจัดการเมลลิงลิสต์ย่อย

1. โปรแกรม Sublist Gateway ซึ่งจะคอยรับจดหมายมาประมวลผล และส่งต่อให้ระหว่างเมลลิงลิสต์หลักและเมลลิงลิสต์ย่อย ดังแสดงในมอดูล (Module) ที่ 1.0 ในรูป Sublist Gateway สามารถแบ่งการทำงานออกเป็นมอดูลย่อยๆ ได้ห้ามอดูลดังนี้

- 1.1. Get Mail
- 1.2. Check for Duplicate Mail by its Message-Id
- 1.3. Check Recipient
- 1.4. Process Mail to Main List
- 1.5. Process Mail to Sublist

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรม System Maintenance ซึ่งจะคอยลบล็อก (Log) ต่างๆ และคอยลบจดหมายคั่นฉบับจากเมลลิงลิสต์หลักที่เก็บไว้ที่แม่ข่ายส่วนกลางเมื่อจดหมายฉบับนั้นหมดอายุ ดังแสดงในมอดูลที่ 2.0 System Maintenance สามารถแบ่งการทำงานออกเป็นมอดูลย่อยๆ ได้สองมอดูลดังนี้

2.1. Delete Expired Message-Id

2.2. Delete Expired Mail

#### 4.3.2 สภาพแวดล้อมของระบบ

ระบบจัดการเมลลิงลิสต์ย่อยถูกออกแบบให้ทำงานบนระบบปฏิบัติการยูนิกซ์ หรือระบบที่เข้ากันได้ (Compatible) เช่น ลินุกซ์ เนื่องจากยูนิกซ์เป็นระบบเปิด ทำให้ง่ายต่อการศึกษา พัฒนา และสะดวกในการปรับแต่ง นอกจากนี้ยูนิกซ์ยังเป็นระบบปฏิบัติการที่เป็นที่นิยมใช้ในการให้บริการอินเทอร์เน็ตมากที่สุด เนื่องจากมีความเชื่อถือได้สูง

เนื่องจากการประมวลผลที่เกี่ยวกับไปรษณีย์อิเล็กทรอนิกส์ จะเป็นการประมวลผลกับข้อความ (Text) เป็นส่วนใหญ่ ดังนั้นภาษาที่เหมาะสมที่จะใช้ในการเขียนโปรแกรมคือภาษา Perl เนื่องจากเป็นภาษาที่มีจุดเด่นในด้านการประมวลผลข้อความ และ Pattern Matching และจะติดต่อกับ Mail Transport Agent ด้วยโพรโทคอล SMTP

ระบบจัดการเมลลิงลิสต์ย่อยจะทำงานโดยอาศัยบัญชีผู้ใช้ (Account) ที่ถูกสร้างขึ้นบนระบบปฏิบัติการยูนิกซ์ และกำหนดให้ Mail Alias ของบัญชีผู้ใช้นี้ให้ส่งจดหมายที่ได้รับต่อให้ Sublist Gateway ผ่านทาง Standard Input จากนั้นก็ใช้บัญชีผู้ใช้นี้สมัครเป็นสมาชิกทั้งเมลลิงลิสต์หลักและเมลลิงลิสต์ย่อย โดยส่งจดหมายไปสมัครสมาชิกตามวิธีที่เมลลิงลิสต์เหล่านั้นกำหนด และบันทึกอีเมลแอดเดรสของเมลลิงลิสต์หลักและเมลลิงลิสต์ย่อยนั้นไว้ในแฟ้มข้อมูลเพื่อให้ Sublist Gateway ใช้เป็นข้อมูลในการส่งต่อจดหมายระหว่างเมลลิงลิสต์ นอกจากนี้ควรเก็บโปรแกรมและแฟ้มข้อมูลที่เกี่ยวข้องต่างๆ ของระบบจัดการเมลลิงลิสต์ย่อยไว้ในไดเรกทอรีเดียวกัน เช่น โฮมไดเรกทอรีของบัญชีผู้ใช้ที่สร้างขึ้น เมื่อมีจดหมายเข้ามาถึงบัญชีผู้ใช้ที่สร้างขึ้นนี้ Sublist Gateway ก็จะถูกเรียกมาทำงานโดยอัตโนมัติผ่านระบบ Mail Alias ที่กำหนด

การเข้าถึงแฟ้มจดหมายคั่นฉบับจากเมลลิงลิสต์หลักสามารถกระทำได้หลายทาง ไม่ว่าจะเป็นการเข้าถึงโดยตรงแบบ Local File สำหรับสมาชิกของเมลลิงลิสต์ย่อยที่อยู่ในเครือข่ายเดียวกับระบบจัดการเมลลิงลิสต์ย่อย (LAN) หรือผ่านบริการต่างๆ เช่น WWW หรือ FTP ซึ่งจะทำให้สมาชิกของเมลลิงลิสต์ย่อยที่อยู่ในเครือข่ายใกล้เคียงสามารถเข้าถึงได้ ขึ้นอยู่กับการกำหนด URL ที่ส่งไปกับจดหมายถึงเมลลิงลิสต์ย่อย

### 4.3.3 รูปแบบของแฟ้มข้อมูล

#### 4.3.3.1 Gateway.ini (Configuration)

Gateway.ini เป็นแฟ้มข้อความที่เก็บค่า Configuration ต่างๆ ของระบบจัดการเมลลิงลิสต์ย่อย มีรูปแบบในการเก็บข้อมูลดังต่อไปนี้

```
[Main]
Parameter1=Value1
Parameter2=Value2
...
```

โดยบรรทัดที่ขึ้นต้นด้วย hash mark (#) หรือ semicolon (;) ใช้แสดงหมายเหตุ (Comment) และโปรแกรมจะไม่สนใจบรรทัดเหล่านี้ พารามิเตอร์แต่ละตัวมีความหมายดังตารางที่ 4.1

ตารางที่ 4.1 แสดงความหมายของพารามิเตอร์แต่ละตัวของแฟ้ม Gateway.ini

พารามิเตอร์	ความหมาย	รูปแบบ
MyAddress	อีเมลแอดเดรสของ Sublist gateway	user@subdomain.subdomain
MailBox	ชื่อแฟ้มของ mailbox ที่ใช้เก็บจดหมายที่ไม่สามารถประมวลผลได้ ถ้าไม่ใส่ path เต็มจะเก็บไว้ในไดเรกทอรีเดียวกับ Sublist gateway	[/path/path/]mailbox
ArchiveDir	ชื่อไดเรกทอรีที่ใช้เก็บจดหมายที่ส่วนกลางสำหรับสมาชิกของเมลลิงลิสต์ย่อย ถ้าไม่ใส่ path เต็มจะเก็บไว้ในไดเรกทอรีเดียวกับ Sublist gateway	[/path/path/]archivedir
URLPrefix	URL ส่วนเดิมนำ เพื่อเข้าถึงไดเรกทอรีที่ใช้เก็บจดหมายที่ส่วนกลาง	Scheme://domain/path
Expire	จำนวนวัน ที่จะเก็บจดหมายที่ส่วนกลาง	N (ตัวเลขแสดงจำนวน)

และต่อไปนี้เป็นตัวอย่างในการเก็บข้อมูลของแฟ้ม Gateway.ini

```
[Main]
MyAddress=sublist@sor.tfrc.co.th
;MailBox=mbox
ArchiveDir=public_html
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

URLPrefix=http://sor.tfrc.co.th/~sublist/  
 Expire=30

#### 4.3.3.2 List.db (Mailing List Address)

List.db เป็นแฟ้มข้อความที่เก็บอีเมลแอดเดรสของเมลลิงลิสต์หลักและเมลลิงลิสต์ย่อยของเมลลิงลิสต์หลักนั้น มีรูปแบบในการเก็บข้อมูลดังต่อไปนี้

```
Local, Remote
Submailinglist address1, Main mailing list address1
Submailinglist address2, Main mailing list address2
...
```

แต่ละฟิลด์ไม่จำกัดความยาว แยกกันด้วยจุดภาค (Comma “,”) และต่อไปนี้เป็นตัวอย่างเป็นการเก็บข้อมูลของแฟ้ม List.db

```
Local, Remote
Sub-lp-1@sor.tfrc.co.th, lp-1@sor.tfrc.co.th
Sub-mj-1@sor.tfrc.co.th, mj-1@sor.tfrc.co.th
```

#### 4.3.3.3 Mid.db (Message-Id Log)

Mid.db เป็นแฟ้มข้อความที่เก็บชื่อของ Message-Id ของจดหมายที่เคยผ่านการประมวลผลมาแล้ว มีรูปแบบในการเก็บข้อมูลดังต่อไปนี้

```
MID, Time
Message_Id1, Timestamp1
Message_Id2, Timestamp2
...
```

แต่ละฟิลด์ไม่จำกัดความยาว แยกกันด้วยจุดภาค (Comma “,”) โดย Timestamp เป็นจำนวนวินาทีตั้งแต่ Epoch (1/1/1970) และต่อไปนี้เป็นตัวอย่างเป็นการเก็บข้อมูลของแฟ้ม Mid.db

```
MID, Time
<Pine.LNX.3.96.980614145703.279A-100000@sor.tfrc.co.th>, 897811073
<Pine.LNX.3.96.980614151055.279B-200000@sor.tfrc.co.th>, 897811953
<Pine.LNX.3.96.1000226170229.449A-100000@sor.tfrc.co.th>, 951559386
```

#### 4.3.3.4 การเก็บจดหมายต้นฉบับจากเมลลิงลิสต์หลักที่ส่วนกลาง

การเก็บจดหมายต้นฉบับจากเมลลิงลิสต์หลักไว้ที่แม่ข่ายส่วนกลาง แยกเก็บจดหมายหนึ่งแฟ้มต่อจดหมายหนึ่งฉบับ โดยใช้ Message-Id ของจดหมายฉบับนั้นเป็นชื่อแฟ้ม และจดหมายจากเมลลิงลิสต์เดียวกันจะเก็บรวมกันในไดเรกทอรีชื่อเดียวกับเมลลิงลิสต์นั้น ภายใต้ไดเรกทอรีที่จะยอมให้สมาชิกของเมลลิงลิสต์ย่อยเข้าถึงได้จากบริการต่างๆ เช่น WWW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

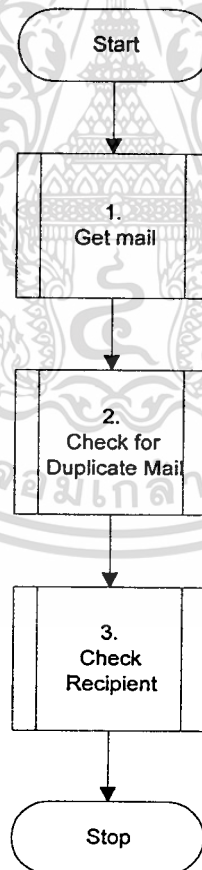
## 4.4 การทำงานของแต่ละมอดูล

การทำงานของมอดูลต่างๆ สามารถอธิบายได้โดยใช้ผังงานโปรแกรม (Program flowchart) ได้ดังต่อไปนี้

### 4.4.1 มอดูล 1.0: Forward Mail between Lists

มอดูลนี้เป็นมอดูลหลักของ Sublist Gateway ซึ่งจะคอยรับจดหมายมาประมวลผล และส่งต่อให้ระหว่างเมลลิงลิสต์หลักและเมลลิงลิสต์ย่อย มีขั้นตอนในการทำงานดังรูปที่ 4.6 ซึ่งแต่ละขั้นตอนสามารถอธิบายได้ดังต่อไปนี้

1. เรียกโปรแกรมย่อย Get mail ขึ้นมาทำงานเพื่อรับจดหมายจาก Standard input มาประมวลผล
2. เรียกโปรแกรมย่อย Check for Duplicate Mail ขึ้นมาทำงานเพื่อตรวจสอบว่าจดหมายที่รับมานั้นเคยผ่านการประมวลผลมาแล้วหรือไม่
3. เรียกโปรแกรมย่อย Check Recipient ขึ้นมาทำงานเพื่อตรวจสอบและประมวลผลจดหมายที่รับมาตามการจำหน่ายถึงผู้รับ

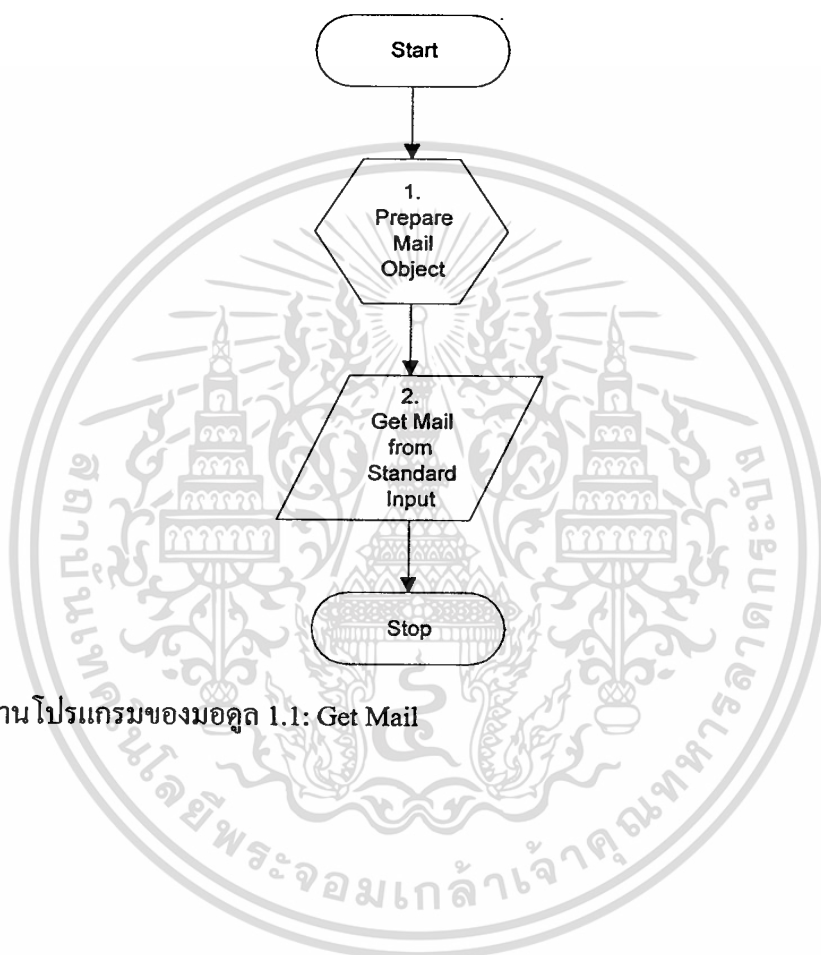


รูปที่ 4.6 ผังงานโปรแกรมของมอดูล 1.0: Forward Mail between Lists

#### 4.4.1.1 มอดูล 1.1: Get Mail

มอดูลนี้ทำหน้าที่รับจดหมายจาก Standard input เข้ามาเก็บไว้ในหน่วยความจำเพื่อใช้ประมวลผลต่อไป ซึ่งมีขั้นตอนในการทำงานดังผังงาน โปรแกรมในรูปที่ 4.7 และคำอธิบายดังนี้

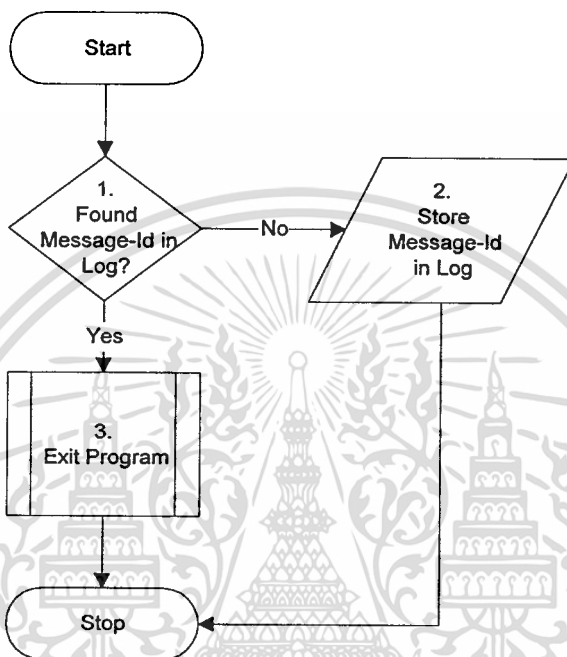
1. เตรียมหน่วยความจำไว้รองรับจดหมายที่จะรับเข้ามา
2. รับจดหมายจาก Standard input เข้ามาเก็บไว้ในหน่วยความจำที่เตรียมไว้เพื่อให้มอดูลอื่นประมวลผลต่อไป



รูปที่ 4.7 ผังงาน โปรแกรมของมอดูล 1.1: Get Mail

#### 4.4.1.2 มอดูล 1.2: Check for Duplicate Mail by its Message-Id

มอดูลนี้ทำหน้าที่ตรวจสอบว่าจดหมายที่รับเข้ามานั้นเคยผ่านการประมวลผลมาแล้วหรือยัง ถ้าหากเคยประมวลผลแล้วก็จะไม่ทำการประมวลผลอีกเพื่อป้องกันการเกิด Mailing loop โดยตรวจสอบ Message-Id ของจดหมายฉบับนั้นกับ Message-Id ของจดหมายที่เคยผ่านการประมวลผลมาแล้ว ซึ่งแสดงเป็นผังงาน โปรแกรมดังรูปที่ 4.8 และคำอธิบายดังนี้

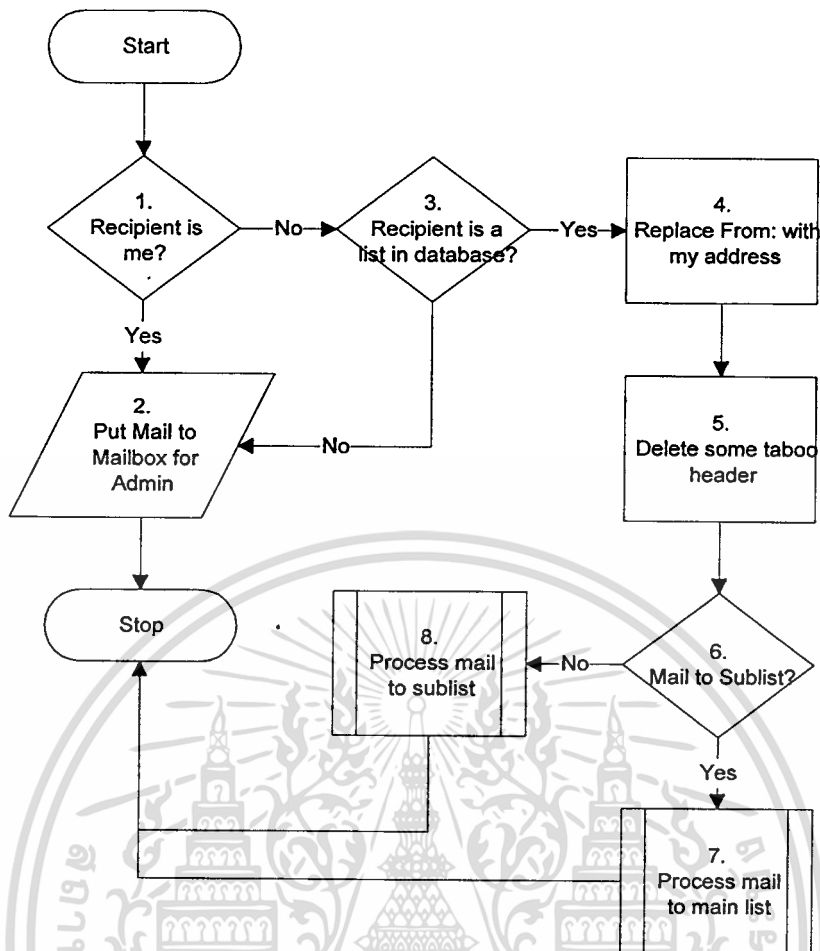


รูปที่ 4.8 ผังงาน โปรแกรมของมอดูล 1.2: Check for Duplicate Mail by its Message-Id

1. ค้นหา Message-Id ของจดหมายที่รับมาเปรียบเทียบกับ Message-Id ของจดหมายที่เคยผ่านการประมวลผลแล้ว หากพบให้ทำต่อในขั้นที่ 3. แต่ถ้าไม่พบให้ทำต่อในขั้นที่ 2.
2. เก็บ Message-Id ของจดหมายฉบับนี้ไว้ในล็อกเพื่อตรวจสอบกับจดหมายฉบับต่อไป
3. ออกจากการทำงานของโปรแกรมเนื่องจากจดหมายฉบับนี้เคยผ่านการประมวลผลมาแล้ว เพื่อไม่ให้เกิด Mailing loop

#### 4.4.1.3 มอดูล 1.3: Check Recipient

มอดูลนี้จะทำการตรวจสอบการจำหน่ายถึงอีเมลแอดเดรสของผู้รับว่าเป็นจดหมายที่จะต้องทำการส่งต่อให้เมลลิงลิสต์หลักหรือเมลลิงลิสต์ย่อยหรือไม่ แสดงได้ดังรูปที่ 4.9 และมีขั้นตอนในการทำงานดังต่อไปนี้



รูปที่ 4.9 ผังงานโปรแกรมของมอดูล 1.3: Check Recipient

1. ตรวจสอบว่าจดหมายฉบับนี้จำหน่ายถึง Sublist gateway เองหรือไม่ ถ้าใช่แสดงว่าไม่ใช่จดหมายที่ต้องทำการส่งต่อให้เมลลิงลิสต์ แต่เป็นจดหมายที่ควรส่งให้ผู้ดูแลระบบจัดการ ให้ทำในขั้นที่ 2. ต่อไป แต่ถ้าไม่ใช่ให้ทำในขั้นที่ 3. ต่อไป
2. เก็บจดหมายฉบับนี้ลงไว้ใน Mailbox เพื่อให้ผู้ดูแลระบบมาจัดการต่อไป
3. ตรวจสอบกับเมลลิงลิสต์แอดเดรสที่ได้ทำการบันทึกไว้ ว่าจดหมายฉบับนี้จำหน่ายถึงผู้รับที่เป็นเมลลิงลิสต์หลักหรือเมลลิงลิสต์ย่อยที่จะต้องทำการส่งต่อให้หรือไม่ ถ้าใช่ให้ประมวลผลต่อไปในขั้นที่ 4. ถ้าไม่ใช่ให้ทำในขั้นที่ 2.
4. ทำการเปลี่ยน Header "From:" เป็นอีเมลแอดเดรสของ Sublist gateway เพื่อให้เมลลิงลิสต์ที่จะส่งต่อไปให้ยอมรับว่าเป็นจดหมายของสมาชิกของเมลลิงลิสต์นั้น
5. ลบ Header ที่อาจทำให้ไม่สามารถส่งต่อจดหมายให้อีกเมลลิงลิสต์ได้ อันได้แก่ Errors-To Precedence และ Header ที่เกี่ยวกับโปรโตคอล MIME
6. ตรวจสอบว่าจดหมายฉบับนี้เป็นจดหมายที่ส่งให้เมลลิงลิสต์ย่อยหรือไม่ ถ้าใช่ให้ทำในขั้นตอนที่ 7. ถ้าไม่ใช่ให้ทำในขั้นตอนที่ 8.

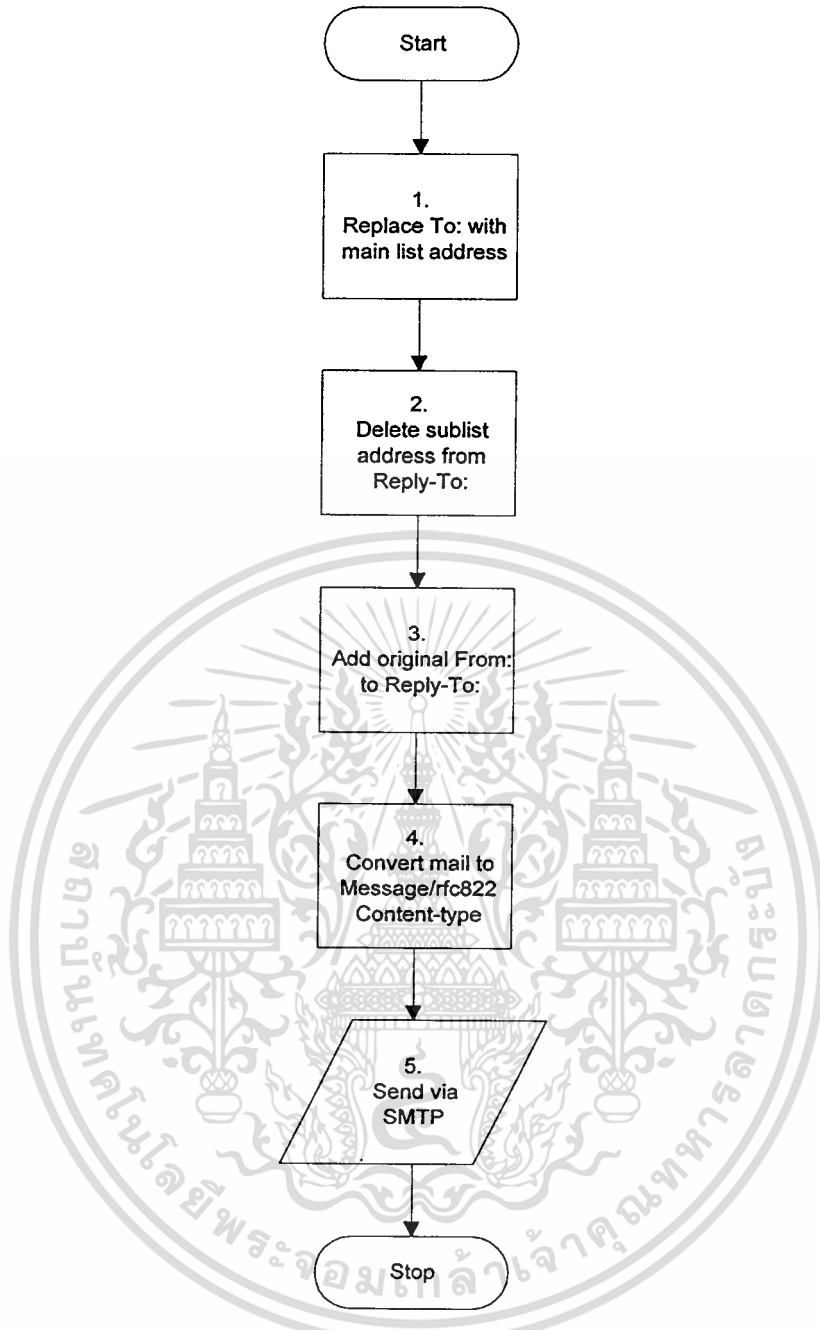
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ทำการส่งจดหมายฉบับนี้ต่อให้เมลลิงลิสต์หลัก โดยเรียกโปรแกรมย่อย Process mail to main list มาทำงาน
8. ทำการส่งจดหมายฉบับนี้ต่อให้เมลลิงลิสต์ย่อย โดยเรียกโปรแกรมย่อย Process mail to sublist มาทำงาน

#### 4.4.1.4 มอดูล 1.4: Process Mail to Main List

มอดูลนี้จะทำการส่งต่อจดหมายให้เมลลิงลิสต์หลัก โดยแก้ไขจดหมายให้เป็นไปตามรูปแบบ Message/rfc822 Content-type ของโปรโทคอล MIME เพื่อรักษา Header ของจดหมายต้นฉบับจากเมลลิงลิสต์ย่อย ไม่ให้ได้รับผลกระทบจากการเปลี่ยนแปลง Header เพื่อให้เมลลิงลิสต์หลักยอมรับว่าจดหมายฉบับนั้นเป็นของสมาชิกของเมลลิงลิสต์หลัก แสดงดังรูปที่ 4.10 และแต่ละขั้นตอนอธิบายการทำงานได้ดังต่อไปนี้

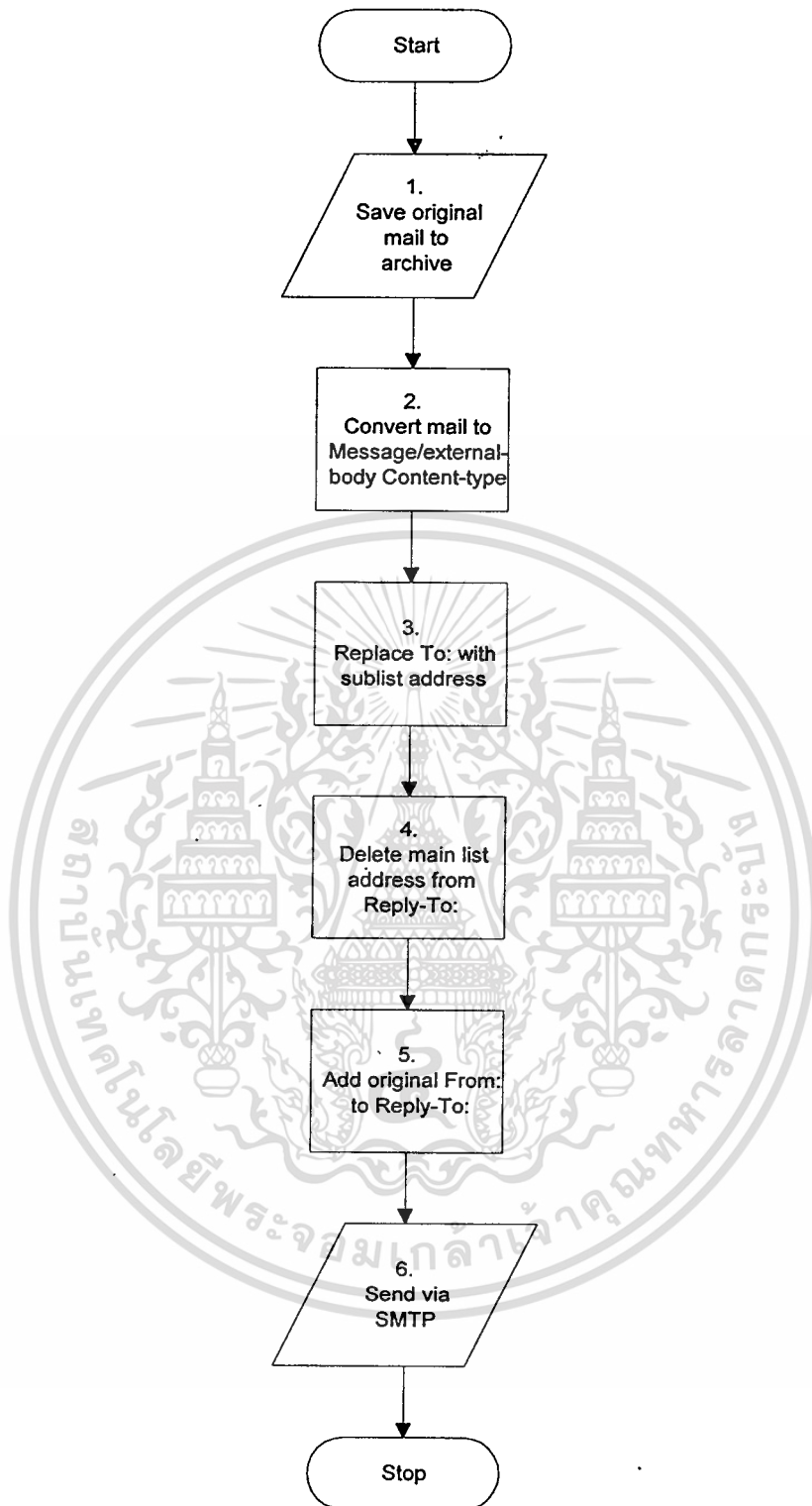
1. เปลี่ยนแปลง Header “To” เป็นอีเมลแอดเดรสของเมลลิงลิสต์หลัก
2. ลบอีเมลแอดเดรสของเมลลิงลิสต์ย่อยออกจาก Header “Reply-To” ถ้ามี เนื่องจากสมาชิกของเมลลิงลิสต์หลักจะไม่มีสิทธิส่งจดหมายตอบให้เมลลิงลิสต์ย่อยอยู่แล้ว เนื่องจากไม่ใช่สมาชิกของเมลลิงลิสต์ย่อย
3. เพิ่มอีเมลแอดเดรสของผู้ส่งจดหมายต้นฉบับ (จาก Header “From:” ของจดหมายต้นฉบับ) ลงไปใน Header “Reply-To” เพื่อให้สมาชิกของเมลลิงลิสต์หลักสามารถตอบจดหมายถึงผู้ส่งที่แท้จริงได้
4. ทำการเปลี่ยนแปลงจดหมายฉบับนั้นให้เป็นไปตามรูปแบบ Message/rfc822 Content-type ของโปรโทคอล MIME
5. ทำการส่งจดหมายที่แก้ไขแล้วให้เมลลิงลิสต์หลักต่อไป โดยใช้โปรโทคอล SMTP ในการติดต่อกับ Mail Transport Agent



รูปที่ 4.10 ผังงาน โปรแกรมของมอดูล 1.4: Process Mail to Main List

#### 4.4.1.5 มอดูล 1.5: Process Mail to Sublist

มอดูลนี้จะทำการส่งต่อจดหมายให้เมลลิงลิสต์ย่อยโดยแก้ไขจดหมายให้เป็นไปตามรูปแบบ Message/external-body Content-type ของโปรโตคอล MIME และเก็บจดหมายต้นฉบับจากเมลลิงลิสต์หลักไว้ที่แม่ข่ายส่วนกลางเพื่อให้สมาชิกของเมลลิงลิสต์ย่อยดึงไปอ่านต่อไป แสดงดังรูปที่ 4.11 และแต่ละขั้นตอนอธิบายการทำงานได้ดังต่อไปนี้



รูปที่ 4.11 ผังงานโปรแกรมของมอดูล 1.5: Process Mail to Sublist

1. เก็บจดหมายต้นฉบับจากเมลลิงลิสต์หลักนั้นไว้ที่แม่ข่ายส่วนกลาง โดยแยกตามไคเร็คทอรี (Directory) ที่ตั้งชื่อตามอีเมลแอดเดรสของเมลลิงลิสต์หลัก และเก็บลงในแฟ้มข้อมูลที่ตั้งชื่อตาม Message-Id ของจดหมายฉบับนั้น โดยแยกเก็บแฟ้มละหนึ่งฉบับ

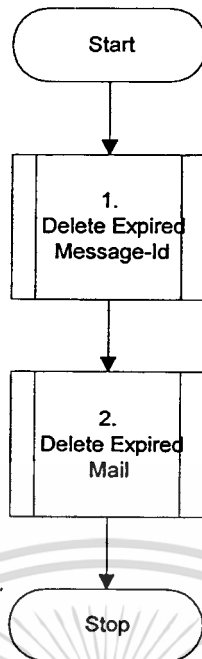
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการเปลี่ยนแปลงจดหมายฉบับนั้นให้เป็นไปตามรูปแบบ Message/external-body Content-type ของโปรโตคอล MIME
3. เปลี่ยนแปลง Header “To” เป็นอีเมลแอดเดรสของเมลลิงลิสต์ย่อย
4. ลบอีเมลแอดเดรสของเมลลิงลิสต์หลักออกจาก Header “Reply-To” ถ้ามี เนื่องจากสมาชิกของเมลลิงลิสต์ย่อยจะไม่มีสิทธิส่งจดหมายตอบให้เมลลิงลิสต์หลักอยู่แล้ว เนื่องจากไม่ใช่สมาชิกของเมลลิงลิสต์หลัก
5. เพิ่มอีเมลแอดเดรสของผู้ส่งจดหมายต้นฉบับ (จาก Header “From:” ของจดหมายต้นฉบับ) ลงไปใน Header “Reply-To” เพื่อให้สมาชิกของเมลลิงลิสต์ย่อยสามารถตอบจดหมายถึงผู้ส่งที่แท้จริงได้
6. ทำการส่งจดหมายที่แก้ไขแล้วให้เมลลิงลิสต์ย่อยต่อไป โดยใช้โปรโตคอล SMTP ในการติดต่อกับ Mail transport agent

#### 4.4.2 มอดูล 2.0: Maintain System

มอดูลนี้เป็นมอดูลหลักของโปรแกรมส่วน System Maintenance ซึ่งจะคอยลบเพิ่มข้อมูลที่เกี่ยวข้องจดหมายต้นฉบับจากเมลลิงลิสต์หลักที่เก็บไว้ที่แม่ข่ายส่วนกลาง และลบล็อกต่างๆ ที่ระบบจัดการเมลลิงลิสต์ย่อยบันทึกไว้ เมื่อถึงเวลาที่กำหนด เพื่อคืนเนื้อที่ของแม่ข่ายให้ใช้ประโยชน์อื่นๆ ต่อไป แสดงผังงานโปรแกรมได้ดังรูปที่ 4.12 และแต่ละขั้นตอนในการทำงานอธิบายได้ดังต่อไปนี้

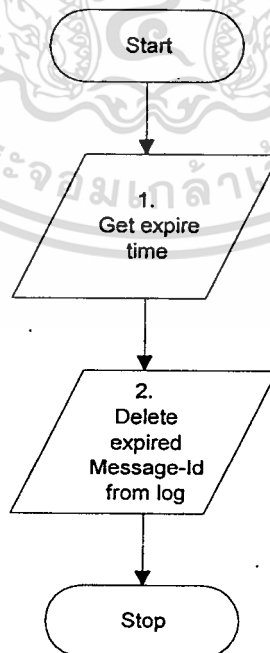
1. เรียกโปรแกรมย่อย Delete Expired Message-Id ขึ้นมาทำงาน เพื่อลบข้อมูลล็อกของ Message-Id ของจดหมายที่เคยผ่านการประมวลผลมาแล้ว เมื่อถึงเวลาที่กำหนด
2. เรียกโปรแกรมย่อย Delete Expired Mail ขึ้นมาทำงาน เพื่อลบเพิ่มข้อมูลที่เกี่ยวข้องจดหมายต้นฉบับเก่าๆ จากเมลลิงลิสต์หลักที่แม่ข่ายส่วนกลาง เมื่อถึงเวลาที่กำหนด



รูปที่ 4.12 ผังงาน โปรแกรมของมอดูล 2.0: Maintain System

#### 4.4.2.1 มอดูล 2.1: Delete Expired Message-Id

มอดูลนี้จะทำหน้าที่ลบชื่อของ Message-Id ของจดหมายที่เคยผ่านการประมวลผลมาแล้วจากเพิ่มข้อมูลที่เก็บ เมื่อถึงเวลาที่กำหนด แสดงเป็นผังงานโปรแกรมได้ดังรูปที่ 4.13 และแต่ละขั้นตอนสามารถอธิบายได้ดังต่อไปนี้



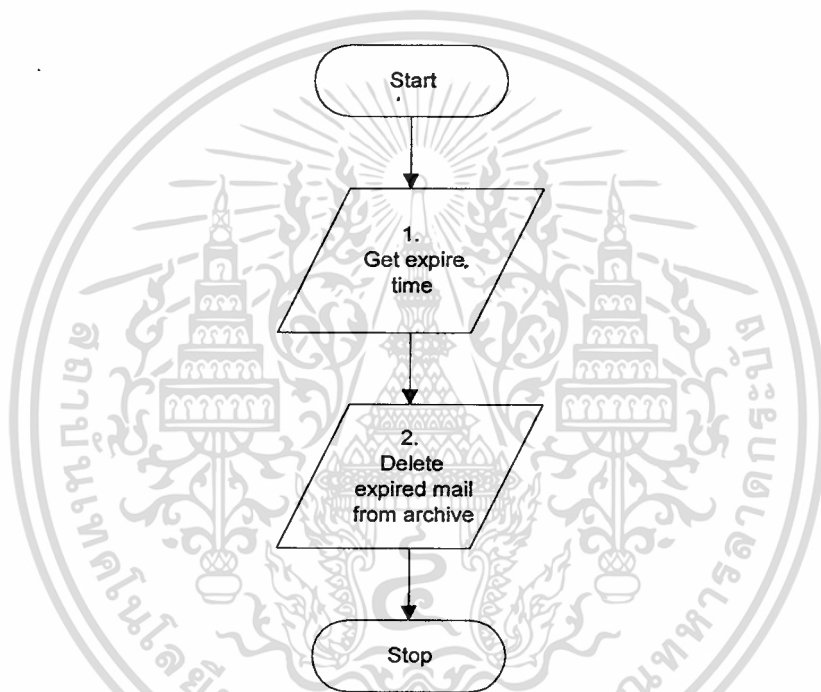
รูปที่ 4.13 ผังงาน โปรแกรมของมอดูล 2.1: Delete Expired Message-Id

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. อ่านค่าระยะเวลา (วัน) ที่กำหนดให้เก็บ Message-Id เก่าของจดหมายที่เคยผ่านการประมวลผลมาแล้วไว้จากเพิ่มข้อมูล Configuration
2. ลบระเบียน (Record) ของ Message-Id ที่มี Timestamp เก่าเกินกว่าระยะเวลาที่อ่านเข้ามาจากขั้นที่ 1 จากเพิ่มข้อมูลที่เก็บชื่อของ Message-Id

#### 4.4.2.2 มอดูล 2.2: Delete Expired Mail

มอดูลนี้จะทำหน้าที่ลบเพิ่มข้อมูลที่เก็บจดหมายต้นฉบับจากเมลลิงลิสต์หลักที่แม่ข่ายส่วนกลาง เมื่อถึงเวลาที่กำหนด แสดงเป็นผังงานโปรแกรมได้ดังรูปที่ 4.14 และแต่ละขั้นตอนสามารถอธิบายได้ดังต่อไปนี้



รูปที่ 4.14 ผังงานโปรแกรมของมอดูล 2.2: Delete Expired Mail

1. อ่านค่าระยะเวลา (วัน) ที่กำหนดให้เก็บจดหมายต้นฉบับจากเมลลิงลิสต์หลักที่แม่ข่ายส่วนกลาง จากเพิ่มข้อมูล Configuration
2. ลบเพิ่มข้อมูลที่เก็บจดหมายต้นฉบับจากเมลลิงลิสต์หลักที่แม่ข่ายส่วนกลาง ที่มีอายุของเพิ่มเกินกว่าระยะเวลาที่อ่านเข้ามาจากขั้นที่ 1

## บทที่ 5

### ผลการทดลอง

#### 5.1 การออกแบบการทดลอง

การออกแบบการทดลองเพื่อวัดปริมาณข้อมูลที่ส่งระหว่างเครือข่าย และเนื้อที่ที่ใช้ในการเก็บจดหมายจากเมลลิงลิสต์ ทำได้โดยการทดลองสร้างเมลลิงลิสต์ย่อยของเมลลิงลิสต์ที่ใช้งานอยู่จริงสามลิสต์ เพื่อเปรียบเทียบความแตกต่างของแต่ละเมลลิงลิสต์ แล้วทำการวัดขนาดของ mailbox ทารด้วยจำนวนจดหมายทั้งหมดเพื่อให้ได้ค่าเฉลี่ยของปริมาณข้อมูลต่อจดหมายหนึ่งฉบับของเมลลิงลิสต์ย่อยที่มีจำนวนสมาชิกตั้งแต่ 1 ถึง 10 คน เปรียบเทียบกับแบบที่ไม่ใช้ระบบจัดการเมลลิงลิสต์ย่อย

#### 5.2 สภาพแวดล้อมของการทดลอง

ในส่วนของฮาร์ดแวร์ประกอบด้วย

- IBM PC Compatible ใช้ CPU Pentium 150 Mhz
- RAM ขนาด 40 MB Hard Disk ขนาด 1.2 GB
- เชื่อมต่อกับอินเทอร์เน็ตด้วย Ethernet LAN Card 10 Mb/s

ในส่วนของซอฟต์แวร์ประกอบด้วย

- ระบบปฏิบัติการลินุกซ์ (Kernel 2.0.29)
- ใช้ BSD sendmail 8.8.5 เป็น Mail Transport Agent
- ใช้ Majordomo 1.94.3 เป็นซอฟต์แวร์ในการจัดการเมลลิงลิสต์

เมลลิงลิสต์ที่ใช้ในการทดลองคือ

- Communication Network (commnet@nectec.or.th) ทดลองกับจดหมาย 160 ฉบับ
- List Managers (list-managers@greatcircle.com) ทดลองกับจดหมาย 294 ฉบับ
- Majordomo Users (majordomo-users@greatcircle.com) ทดลองกับจดหมาย 1999 ฉบับ

#### 5.3 การทดลองปริมาณข้อมูลที่ส่งระหว่างเครือข่าย

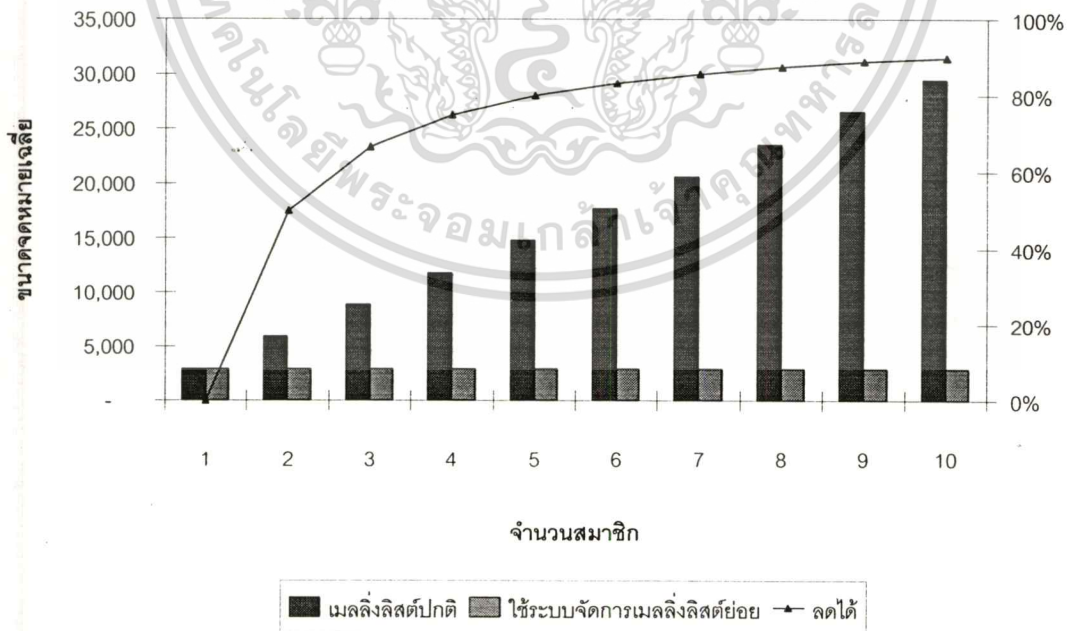
##### 5.3.1 ผลการทดลอง

ตารางต่อไปนี้จะแสดงปริมาณข้อมูลเฉลี่ยต่อฉบับของจดหมายที่ส่งระหว่างเครือข่าย เปรียบเทียบกันระหว่างเมลลิงลิสต์ที่ใช้และไม่ใช้ระบบจัดการเมลลิงลิสต์ย่อย

ตารางที่ 5.1 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ Communication Network

จำนวนสมาชิก (คน)	เมลลิงลิสต์ปกติ (ไบต์/ฉบับ)	เมลลิงลิสต์ที่ใช้ระบบจัดการ เมลลิงลิสต์ย่อย (ไบต์/ฉบับ)	ลดปริมาณ ของข้อมูลได้
1	2,948	2,948	0%
2	5,896	2,948	50%
3	8,844	2,948	67%
4	11,792	2,948	75%
5	14,740	2,948	80%
6	17,688	2,948	83%
7	20,636	2,948	86%
8	23,584	2,948	88%
9	26,532	2,948	89%
10	29,480	2,948	90%

จากตารางที่ 5.1 ข้างต้นสามารถแสดงเป็นกราฟได้ดังรูปที่ 5.1 ต่อไปนี้



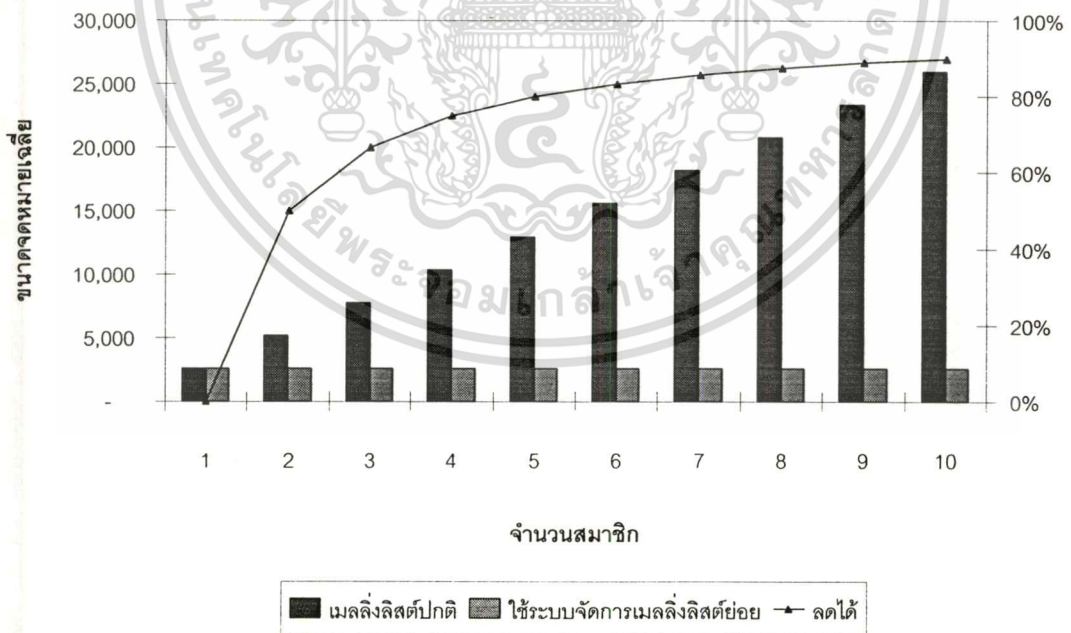
รูปที่ 5.1 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ Communication Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ List Managers

จำนวนสมาชิก (คน)	เมลลิงลิสต์ปกติ (ไบต์/ฉบับ)	เมลลิงลิสต์ที่ใช้ระบบจัดการ เมลลิงลิสต์ย่อย (ไบต์/ฉบับ)	ลดปริมาณ ของข้อมูลได้
1	2,597	2,597	0%
2	5,194	2,597	50%
3	7,791	2,597	67%
4	10,388	2,597	75%
5	12,985	2,597	80%
6	15,582	2,597	83%
7	18,179	2,597	86%
8	20,776	2,597	88%
9	23,373	2,597	89%
10	25,970	2,597	90%

จากตารางข้างต้นสามารถแสดงเป็นกราฟได้ดังรูปต่อไปนี้



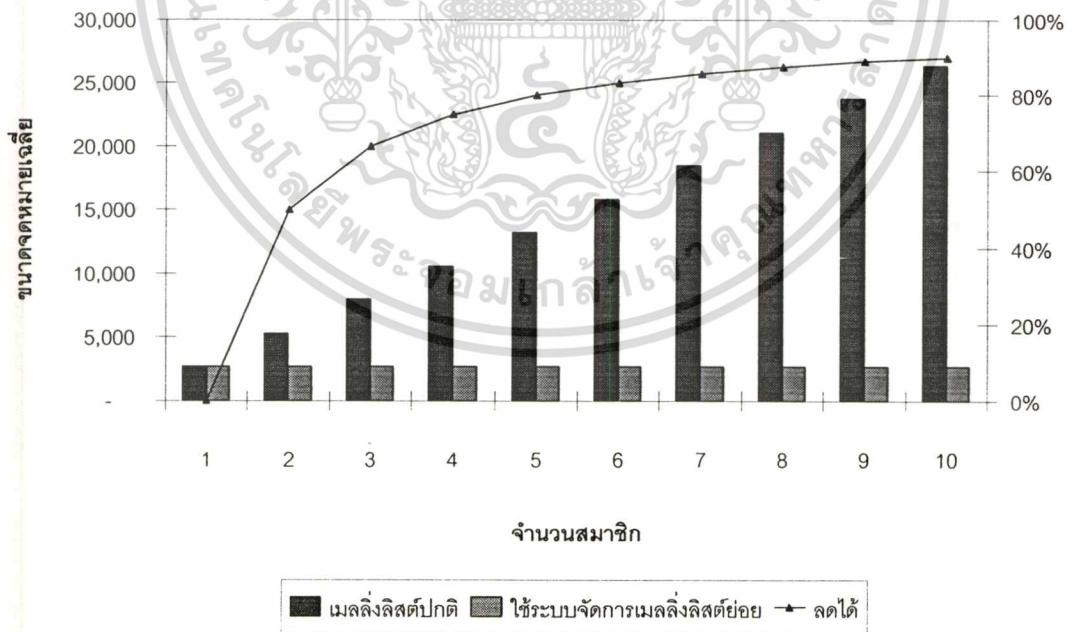
รูปที่ 5.2 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ List Managers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.3 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ Majordomo Users

จำนวนสมาชิก (คน)	เมลลิงลิสต์ปกติ (ไบต์/ฉบับ)	เมลลิงลิสต์ที่ใช้ระบบจัดการ เมลลิงลิสต์ย่อย (ไบต์/ฉบับ)	ลดปริมาณ ของข้อมูลได้
1	2,637	2,637	0%
2	5,274	2,637	50%
3	7,911	2,637	67%
4	10,548	2,637	75%
5	13,185	2,637	80%
6	15,822	2,637	83%
7	18,459	2,637	86%
8	21,096	2,637	88%
9	23,733	2,637	89%
10	26,370	2,637	90%

จากตารางข้างต้นสามารถแสดงเป็นกราฟได้ดังรูปต่อไปนี้



รูปที่ 5.3 ปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์ Majordomo Users

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3.2 วิเคราะห์ผลการทดลอง

จากผลการทดลอง แสดงการเปรียบเทียบให้เห็นได้ว่า ไม่ว่าจะภายในเครือข่ายเดียวกันหรือใกล้เคียงกันจะมีกลุ่มสมาชิกกี่คน หากใช้ระบบจัดการเมลลิงลิสต์ย่อยปริมาณข้อมูลโดยเฉลี่ยของจดหมายแต่ละฉบับจะคงที่ไม่เพิ่มขึ้นตามจำนวนสมาชิกเนื่องจากจะมีการส่งจดหมายมาเพียงฉบับเดียวต่อครั้งแทนที่จะทำสำเนาจดหมายเท่าจำนวนสมาชิกในลิสต์ย่อยแล้วส่งมาตามแบบเดิม

ดังนั้นยังสามารถรวบรวมสมาชิกเป็นลิสต์ย่อยได้มากเท่าไร ก็ยิ่งลดปริมาณจดหมายที่ต้องส่งมาจากเมลลิงลิสต์หลักได้มากเท่านั้น

จากผลการทดลอง สามารถแสดงความสัมพันธ์ของปริมาณข้อมูลที่ลดได้เมื่อใช้เมลลิงลิสต์ย่อยเทียบเป็นอัตราส่วนกับแบบไม่ใช้ ( $P_n$ ) เมื่อมีสมาชิก  $n$  คนเมื่อ  $M_n$  แทนปริมาณข้อมูลที่ส่งระหว่างเครือข่ายของเมลลิงลิสต์แบบปกติ และ  $L_n$  แทนแบบใช้เมลลิงลิสต์ย่อย ได้ดังต่อไปนี้

$$P_n = \frac{(M_n - L_n)}{M_n} \quad (1)$$

$$P_n = \frac{(nL_n - L_n)}{nL_n} \quad (2)$$

$$P_n = 1 - \frac{1}{n} \quad (3)$$

จาก (3) แสดงให้เห็นว่า ระบบจัดการเมลลิงลิสต์ย่อยจะลดปริมาณข้อมูลที่ส่งระหว่างเครือข่ายได้เมื่อใช้กับเมลลิงลิสต์ย่อยที่มีสมาชิก ( $n$ ) ตั้งแต่สองคนขึ้นไปเนื่องจากถ้า  $n$  เป็นหนึ่งจะทำให้  $P_n$  มีค่าเป็นศูนย์ และยังมีสมาชิกมากเท่าไร จะยิ่งลดปริมาณข้อมูลที่ส่งระหว่างเครือข่าย ( $P_n$ ) ได้มากเท่านั้นเนื่องจากพจน์  $\frac{1}{n}$  จะยังมีค่าน้อยลง

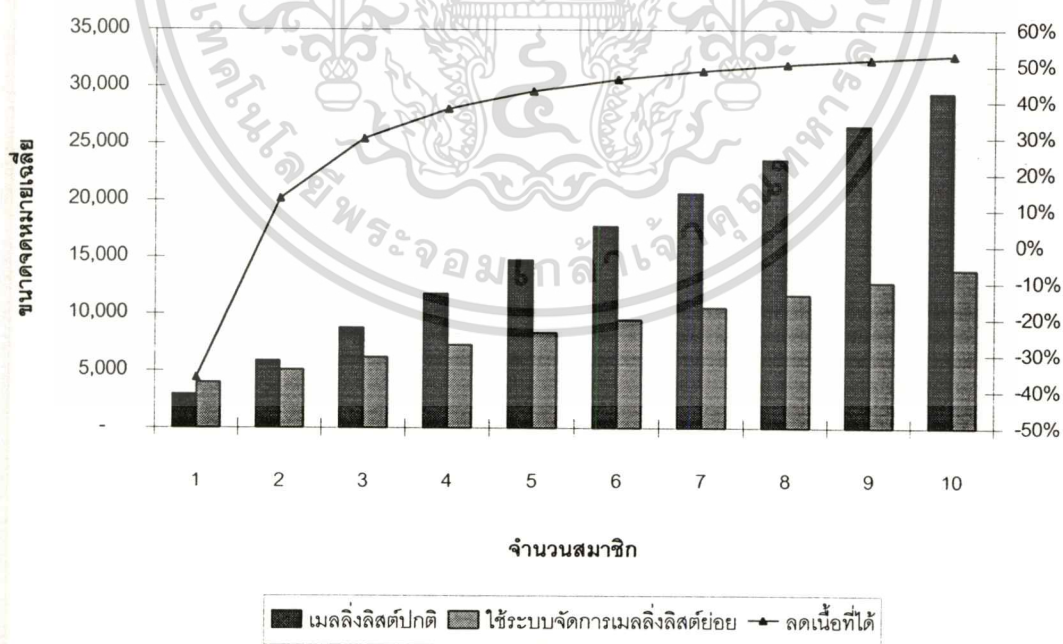
## 5.4 การทดลองเนื้อที่ที่ใช้ในการเก็บจดหมาย

### 5.4.1 ผลการทดลอง

จากการทดลองสามารถวัดขนาดของ Mailbox ที่ใช้เก็บจดหมายของกลุ่มสมาชิกเพื่อหาขนาดเฉลี่ยต่อฉบับของจดหมาย เปรียบเทียบระหว่างเมลลิงลิสต์แบบปกติที่ไม่ได้ใช้ระบบจัดการเมลลิงลิสต์ย่อยกับแบบที่ใช้ระบบจัดการเมลลิงลิสต์ย่อยได้ดังตารางและกราฟได้ดังต่อไปนี้

ตารางที่ 5.4 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ Communication Network

จำนวนสมาชิก (คน)	เมลลิงลิสต์ปกติ (ไบต์/ฉบับ)	เมื่อใช้ระบบจัดการเมลลิงลิสต์ ย่อย (ไบต์/ฉบับ)	ลดเนื้อที่ได้
1	2,948	4,012	-36%
2	5,896	5,109	13%
3	8,844	6,206	30%
4	11,792	7,303	38%
5	14,740	8,400	43%
6	17,688	9,497	46%
7	20,636	10,594	49%
8	23,584	11,691	50%
9	26,532	12,788	52%
10	29,480	13,885	53%

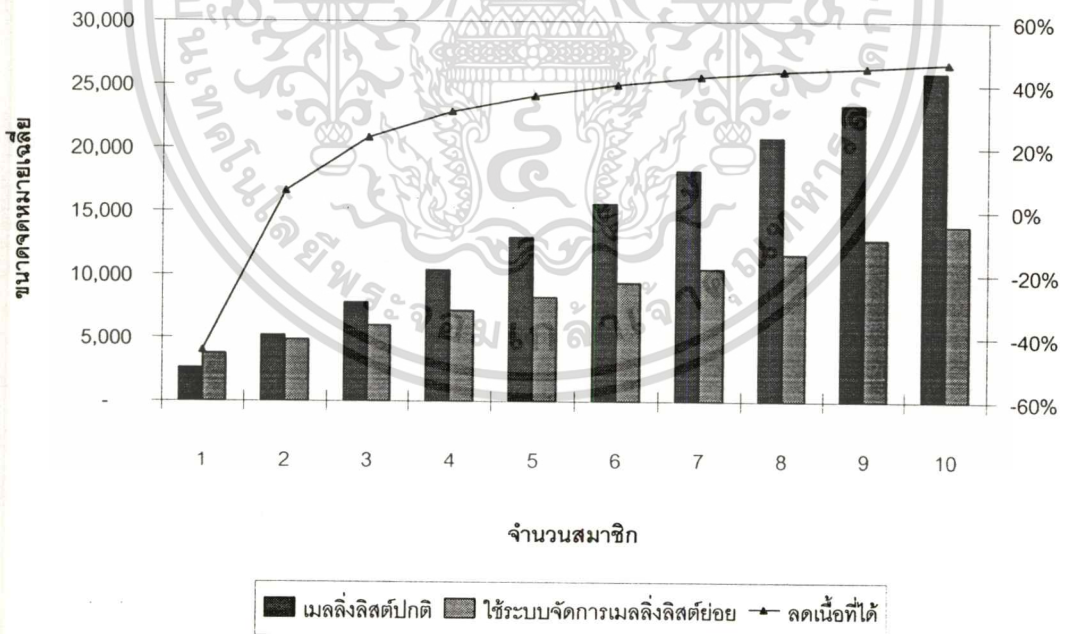


รูปที่ 5.4 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ Communication Network

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.5 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ List Managers

จำนวนสมาชิก (คน)	เมลลิงลิสต์ปกติ (ไบต์/ฉบับ)	เมื่อใช้ระบบจัดการเมลลิงลิสต์ ย่อย (ไบต์/ฉบับ)	ลดเนื้อที่ได้
1	2,597	3,734	-44%
2	5,194	4,859	6%
3	7,791	5,984	23%
4	10,388	7,109	32%
5	12,985	8,234	37%
6	15,582	9,359	40%
7	18,179	10,484	42%
8	20,776	11,609	44%
9	23,373	12,734	46%
10	25,970	13,859	47%

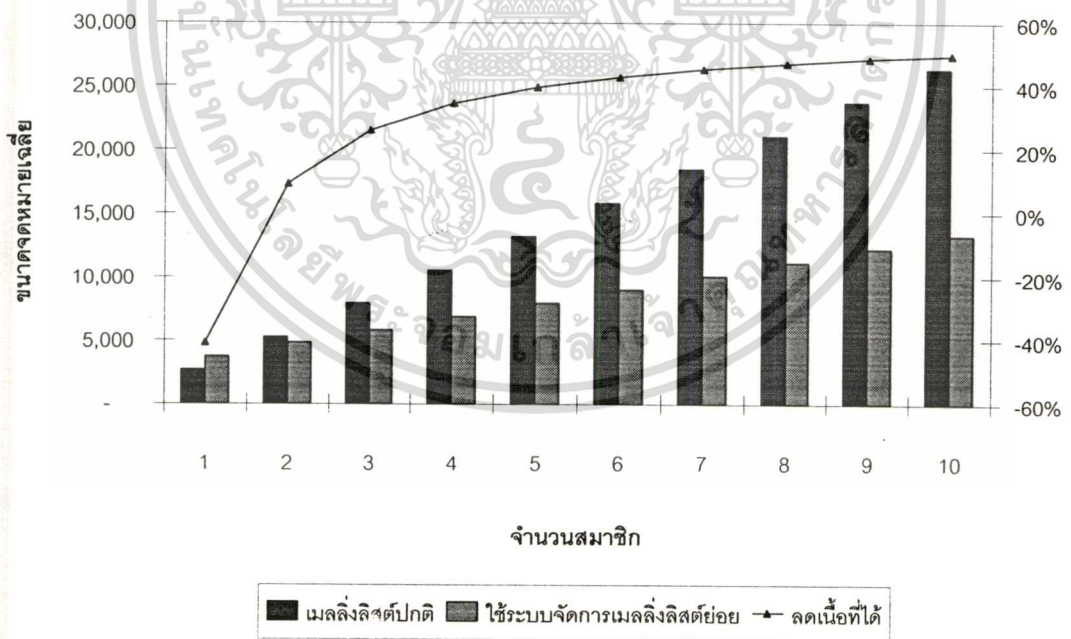


รูปที่ 5.5 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ List Managers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

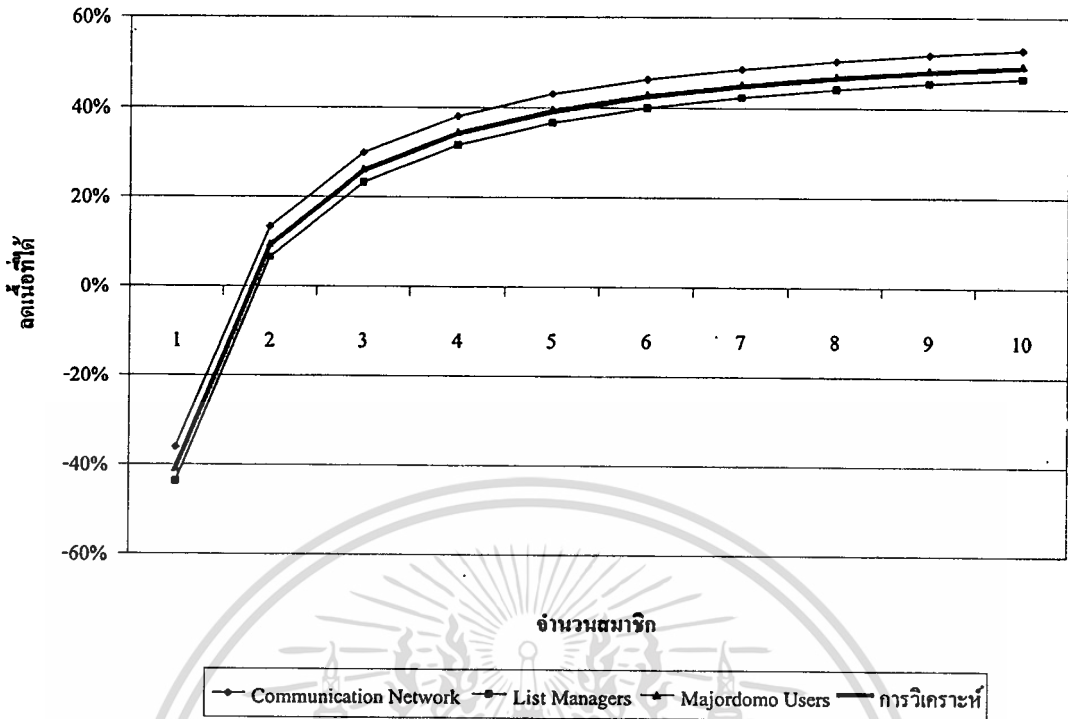
ตารางที่ 5.6 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ Majordomo Users

จำนวนสมาชิก (คน)	เมลลิงลิสต์ปกติ (ไบต์/ฉบับ)	เมื่อใช้ระบบจัดการเมลลิงลิสต์ ย่อย (ไบต์/ฉบับ)	ลดเนื้อที่ได้
1	2,637	3,714	-41%
2	5,274	4,779	9%
3	7,911	5,844	26%
4	10,548	6,909	34%
5	13,185	7,974	40%
6	15,822	9,039	43%
7	18,459	10,104	45%
8	21,096	11,169	47%
9	23,733	12,234	48%
10	26,370	13,299	50%



รูปที่ 5.6 เปรียบเทียบเนื้อที่เฉลี่ยที่ใช้เก็บจดหมายของเมลลิงลิสต์ Majordomo Users

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.7 เปรียบเทียบเนื้อที่ที่ลดได้จากการทดลองกับเมลลิงลิสต์ทั้งสามและการวิเคราะห์

#### 5.4.2 วิเคราะห์ผลการทดลอง

จากผลการทดลองที่แสดง เมื่อใช้ระบบจัดการเมลลิงลิสต์ย่อย ในสมาชิกแต่ละคนจะเสียเนื้อที่ในการเก็บจดหมายประมาณ 1 กิโลไบต์ต่อฉบับไม่ว่าจดหมายต้นฉบับจะมีขนาดเท่าไรก็ตาม เนื่องจากจดหมายที่ส่งจากระบบจัดการเมลลิงลิสต์ย่อยให้สมาชิกในลิสต์ย่อยจะเป็นคล้ายตัวชี้ที่บอกวิธีในการเข้าถึงต้นฉบับตัวจริงที่เก็บไว้ที่ส่วนกลางเพียงสำเนาเดียวเท่านั้น ซึ่งสามารถแสดงความสัมพันธ์ของเนื้อที่ที่ลดได้เมื่อใช้เมลลิงลิสต์ย่อยเทียบเป็นอัตราส่วนกับแบบไม่ใช่ ( $P_s$ ) กับจำนวนสมาชิก  $n$  คน ขนาดเฉลี่ยของจดหมายจากเมลลิงลิสต์หลัก ( $s$ ) และขนาดเฉลี่ยของจดหมายจากเมลลิงลิสต์ย่อย ( $h$ ) ได้ดังสมการต่อไปนี้ โดยให้  $M_s$  แทนเนื้อที่ที่ใช้เก็บจดหมายของเมลลิงลิสต์แบบปกติ และให้  $L_s$  แทนแบบใช้เมลลิงลิสต์ย่อย และเมื่อแทนค่า  $h$  และ  $s$  ด้วยค่าเฉลี่ยจากการทดลองกับเมลลิงลิสต์ทั้งสาม สามารถแสดงเป็นกราฟเปรียบเทียบได้ดังรูปที่ 5.7

$$P_s = \frac{(M_s - L_s)}{M_s} \quad (4)$$

$$P_s = \frac{(ns - (nh + s))}{ns} \quad (5)$$

$$P_s = 1 - \frac{h}{s} - \frac{1}{n} \quad (6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก (6) แสดงให้เห็นว่า เมลลิงลิสต์ที่ใช้ควรมีขนาดเฉลี่ยของจดหมาย ( $s$ ) มากกว่าขนาดเฉลี่ยของจดหมายจากเมลลิงลิสต์ย่อย ( $h$ ) ซึ่งจะมีขนาดประมาณ 1 กิโลไบต์ เนื่องจากหากมีขนาดจดหมายเฉลี่ยน้อยกว่า 1 กิโลไบต์ จะทำให้พจน์  $\frac{h}{s}$  มีค่ามากกว่าหนึ่ง ทำให้  $P_s$  ติดลบเสมอไม่ว่าจำนวนสมาชิกของเมลลิงลิสต์ย่อย ( $n$ ) จะเป็นเท่าไร

และสมการ (6) ยังแสดงให้เห็นว่า ควรใช้ระบบจัดการเมลลิงลิสต์ย่อยเมื่อสามารถรวบรวมสมาชิกเป็นลิสต์ย่อยได้ตั้งแต่ 2 คนขึ้นไป เพราะถ้ามีสมาชิกในลิสต์ย่อยเพียงคนเดียวระบบจัดการเมลลิงลิสต์ย่อยกลับจะเพิ่มเนื้อที่ในการเก็บจดหมายเสมอ (จากสมการ  $P_s$  จะติดลบเสมอเมื่อ  $n = 1$ ) เนื่องจากต้องเสียเนื้อที่ในการเก็บจดหมายขนาดประมาณ 1 กิโลไบต์ที่ส่งให้แก่สมาชิกดังที่กล่าวมาเพิ่มเติมจากต้นฉบับเดิม

นอกจากนี้ยังวิเคราะห์ได้ว่า ระบบจัดการเมลลิงลิสต์ย่อยจะยังลดเนื้อที่ในการเก็บจดหมาย ( $P_s$ ) ได้มากขึ้น เพียงแค่ขนาดเฉลี่ยของจดหมาย ( $s$ ) หรือจำนวนสมาชิก ( $n$ ) ตัวใดตัวหนึ่งมีค่ามากขึ้น เช่น จากสมการ (6) หากต้องการให้ลดเนื้อที่ได้ ( $P_s > 0$ ) เมื่อมีสมาชิก  $n$  คน สามารถคำนวณหาขนาดของ  $s$  ที่เหมาะสมได้ดังนี้

$$P_s > 0$$

$$1 - \frac{h}{s} - \frac{1}{n} > 0$$

$$s > \frac{nh}{n-1} \tag{7}$$

$$s > \frac{h}{P_n} \tag{8}$$

ดังนั้น จาก (7) หากต้องการหาขนาดเฉลี่ยของเมลลิงลิสต์ที่เหมาะสมเมื่อมีสมาชิกในลิสต์ย่อยน้อยเพียง 2 คน สามารถแทนค่าได้ดังนี้

$$s > \frac{2 \times 1024}{2-1}$$

$$s > 2048$$

ดังนั้นหากมีสมาชิกในลิสต์ย่อยเพียง 2 คน ควรใช้กับเมลลิงลิสต์ที่มีขนาดจดหมายเฉลี่ยมากกว่า 2 กิโลไบต์ขึ้นไป

## บทที่ 6

# สรุปผลการวิจัยและข้อเสนอแนะ

### 6.1 สรุปผลการวิจัย

จากปัญหาของความซ้ำซ้อนในการส่งจดหมายของเมลลิงลิสต์ถึงสมาชิกที่อยู่ในเครือข่ายเดียวหรือใกล้เคียงกัน ระบบจัดการเมลลิงลิสต์ย่อยแก้ปัญหานี้โดยการแบ่งสมาชิกออกเป็นกลุ่มเมลลิงลิสต์ย่อยตามความใกล้เคียงของเครือข่ายของสมาชิก แต่เนื่องจากซอฟต์แวร์จัดการเมลลิงลิสต์จะไม่ยอมให้ผู้ที่ไม่ใช่สมาชิกส่งจดหมายให้เมลลิงลิสต์ ด้วยเหตุนี้จึงจดหมายจากเมลลิงลิสต์ย่อยจะไม่สามารถส่งถึงเมลลิงลิสต์หลักได้ และยังไม่สามารถใช้ซอฟต์แวร์จัดการเมลลิงลิสต์มาจัดการเมลลิงลิสต์ย่อยอีกด้วย ปัญหานี้แก้โดยการสร้าง Sublist gateway เป็นตัวแลกเปลี่ยนจดหมายระหว่างเมลลิงลิสต์หลัก และ เมลลิงลิสต์ย่อย โดยทำหน้าที่เสมือนเป็นสมาชิกของทั้งสองเมลลิงลิสต์ เมื่อได้รับจดหมายจากเมลลิงลิสต์ใดก็จะทำเสมือนเอาจดหมายฉบับนั้นใส่ลงในช่องใหม่แล้วส่งต่อให้อีกเมลลิงลิสต์หนึ่ง เหมือนเป็นจดหมายที่มาจากสมาชิกของเมลลิงลิสต์ปลายทาง ทำให้เกิดการโต้ตอบกันได้ทั้งสองทาง และสามารถนำซอฟต์แวร์จัดการเมลลิงลิสต์ใดๆ มาจัดการเมลลิงลิสต์ย่อยได้

ส่วนปัญหาของความซ้ำซ้อนกันของเนื้อหาที่ใช้เก็บจดหมายจากเมลลิงลิสต์ ในสมาชิกที่อยู่ในเครือข่ายเดียวหรือใกล้เคียงกัน และการตกค้างของจดหมายเก่าๆ จากเมลลิงลิสต์ สามารถแก้ได้โดยเก็บจดหมายต้นฉบับไว้ที่แม่ข่ายส่วนกลางสำหรับแต่ละเมลลิงลิสต์ย่อย แล้วส่งจดหมายที่เป็นตัวชี้บอกวิธีในการเข้าถึงจดหมายต้นฉบับนั้นให้แก่สมาชิกของเมลลิงลิสต์ย่อย ตามโพรโทคอล MIME ที่มี Content-type แบบ Message/External-body และคอยลบจดหมายต้นฉบับเก่าที่ส่วนกลางทิ้งเป็นระยะๆ

### 6.2 ประโยชน์ของระบบจัดการเมลลิงลิสต์ย่อย

เมื่อนำระบบจัดการเมลลิงลิสต์ย่อยมาใช้ จะก่อให้เกิดประโยชน์ดังนี้

1. เพิ่มประสิทธิภาพในการส่งข้อมูลระหว่างเครือข่าย หากมีการรวบรวมสมาชิกของเมลลิงลิสต์ใดๆ ในเครือข่ายเดียวกันหรือจากเครือข่ายที่ใกล้เคียงกันเป็นลิสต์ย่อย แล้วจัดการด้วยระบบจัดการเมลลิงลิสต์ย่อย จะลดปริมาณการส่งจดหมายของเมลลิงลิสต์จากการส่งจดหมายซ้ำๆ กัน ให้สมาชิกทุกคนเหลือเพียงฉบับเดียวต่อครั้งสำหรับแต่ละลิสต์ย่อย และจะเป็นประโยชน์อย่างยิ่งหากต้องผ่านจุดที่มีการเชื่อมต่อที่ช้าและต้องเสียค่าใช้จ่ายสูง
2. ช่วยกระจายภาระในการจัดการและดูแลเมลลิงลิสต์ที่มีขนาดใหญ่ เนื่องจากการใช้ระบบจัดการเมลลิงลิสต์ย่อยจะเป็นการกระจายจำนวนสมาชิกขนาดใหญ่ออกเป็นกลุ่มย่อยๆ ทำให้ลด

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อใช้ในการศึกษาวิจัยเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมลลิงลิสต์ได้ เพราะการประมวลผลเพื่อแจกจ่ายจดหมายให้สมาชิกของแต่ละเมลลิงลิสต์ย่อย จะประมวลผลแบบขนานพร้อมกันไป ซึ่งจะเป็นการเพิ่มประสิทธิภาพให้เมลลิงลิสต์โดยรวม สามารถแจกจ่ายจดหมายให้สมาชิกทั้งหมดได้รวดเร็วขึ้น

3. ลดเนื้อที่ในการเก็บจดหมายจากเมลลิงลิสต์ที่ซ้ำซ้อนกันของสมาชิกแต่ละคน โดยในแต่ละลิสต์ย่อย ระบบจัดการเมลลิงลิสต์ย่อยจะเก็บจดหมายต้นฉบับไว้ที่ส่วนกลางแทนที่จะแยกเก็บแต่ละสำเนาใน mailbox ของสมาชิกแต่ละคน ทำให้สามารถแบ่งใช้ทรัพยากรของแม่ข่ายได้คุ้มค่ายิ่งขึ้น
4. ลดปริมาณของจดหมายจากเมลลิงลิสต์ที่ตกค้างสะสมอยู่ใน mailbox ของสมาชิกแต่ละคน เนื่องจากระบบจัดการเมลลิงลิสต์ย่อยจะลบจดหมายที่ส่วนกลางทิ้งเมื่อครบระยะเวลาที่กำหนด เพื่อคืนเนื้อที่ของแม่ข่ายที่ใช้เก็บจดหมายเหล่านี้ จึงเป็นการลดปริมาณการตกค้างสะสมของจดหมายเก่าๆ จากเมลลิงลิสต์ของสมาชิกแต่ละคน ได้

### 6.3 ข้อเสนอแนะ

ในการนำระบบจัดการเมลลิงลิสต์ย่อยมาใช้ให้เกิดประโยชน์สูงสุดนั้น มีปัจจัยที่ต้องได้รับการพิจารณาหลายข้อดังต่อไปนี้

1. จำนวนสมาชิกที่รวบรวมกันเป็นลิสต์ย่อย จากการทดลอง ระบบจัดการเมลลิงลิสต์ย่อยจะไม่ช่วยลดปริมาณจดหมายจากเมลลิงลิสต์ที่ส่งระหว่างเครือข่ายและเนื้อที่ของแม่ข่ายที่ใช้ในการเก็บจดหมายจากเมลลิงลิสต์ ถ้าไม่ใช้กับลิสต์ย่อยที่มีสมาชิกตั้งแต่ 2 คนขึ้นไป
2. ความใกล้เคียงของเครือข่ายของสมาชิกที่รวมกันเป็นลิสต์ย่อย สมาชิกของลิสต์ย่อยควรอยู่ภายในเครือข่ายเดียวกันหรือเครือข่ายที่ใกล้เคียงกับแม่ข่ายของระบบจัดการเมลลิงลิสต์ย่อยมากกว่าลิสต์หลัก หากแบ่งสมาชิกทั้งหมดของเมลลิงลิสต์ขนาดใหญ่ออกเป็นลิสต์ย่อยตามความใกล้เคียงของเครือข่ายแล้ว จะทำประสิทธิภาพในการแจกจ่ายจดหมายให้สมาชิกโดยรวมดีขึ้น และช่วยลดเวลาครบวงงาน (Turnaround time) สำหรับเมลลิงลิสต์ขนาดใหญ่ลง
3. การดึงจดหมายต้นฉบับจากส่วนกลางมาแสดงอาจไม่สามารถทำได้ หรือทำได้ไม่ถูกต้องใน Mail user agent บางตัวที่ไม่สนับสนุนโปรโตคอล MIME อย่างสมบูรณ์ โดยเฉพาะอย่างยิ่งถ้า Mail user agent นั้นไม่สนับสนุน Content-type แบบ message/external-body ดังนั้นหากจะนำระบบจัดการเมลลิงลิสต์ย่อยมาใช้ควรพิจารณา Mail user agent ที่ใช้อยู่ด้วย ยกตัวอย่างเช่น mh สามารถใช้ได้เป็นอย่างดีไม่มีปัญหาใดๆ ส่วน Netscape Messenger อาจมีปัญหาในการแสดงผลจดหมายที่เข้ารหัสบ้าง แต่ Microsoft Outlook จะมีปัญหาค่อนข้างมากเนื่องจากไม่สนับสนุน message/external-body

4. ซอฟต์แวร์จัดการเมลลิ่งลิสต์รุ่นเก่าบางตัวเช่น ListProc 6.0c อาจมีปัญหาเกี่ยวกับ MIME Header เช่น พารามิเตอร์ที่ยาวจนต้องแยกเป็นหลายบรรทัดอย่าง message/external-body จะถูก ListProc ตัดเหลือเพียงบรรทัดเดียว ดังนั้นจึงควรหลีกเลี่ยงในการนำมาใช้จัดการลิสต์ย่อย
5. จากการทดลองแสดงให้เห็นว่า ควรใช้ระบบจัดการเมลลิ่งลิสต์ย่อยกับเมลลิ่งลิสต์ที่มีขนาดของจดหมายเฉลี่ยตั้งแต่ 2 กิโลไบต์/ฉบับขึ้นไป เนื่องจากจะทำให้การใช้เนื้อที่ในการเก็บจดหมายจากเมลลิ่งลิสต์มีประสิทธิภาพดีแม้จะมีจำนวนสมาชิกในลิสต์ย่อยน้อยเพียง 2 คน และไม่ควรรู้ใช้กับเมลลิ่งลิสต์ที่มีขนาดของจดหมายเฉลี่ยน้อยกว่า 1 กิโลไบต์/ฉบับ
6. หากต้องการให้สามารถย้ายสมาชิกจากเมลลิ่งลิสต์หลักแบบปกติมาเป็นเมลลิ่งลิสต์ย่อยโดยอัตโนมัติ ควรเพิ่มตัวกรองจดหมาย (Mail Filter) เข้าไป โดยถ้าใช้ sendmail เป็น Mail Transport Agent สามารถเพิ่มเข้าไปได้โดยกำหนดให้ตัวกรองนี้ทำงานแทน local delivery agent เพื่อคอยตรวจสอบจดหมายที่มาจากเมลลิ่งลิสต์หลักว่าถึงผู้ใช้ใด ก็ทำการเพิ่มผู้ใช้คนนั้นให้เป็นสมาชิกของเมลลิ่งลิสต์ย่อย และส่งจดหมายไปบอกเลิกสมาชิกของผู้ใช้คนนั้นกับเมลลิ่งลิสต์หลักโดยอัตโนมัติ



## เอกสารอ้างอิง

- [1] สรวิศ นุญมี. “ระบบจัดการเมลล์ถึงลิสต์ย่อย.” บทความเสนอในการสัมมนาวิชาสัมมนา 2 ระดับปริญญาโท สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2537.
- [2] Bryan Costales, Eric Allman, Neil Rickert. **sendmail**. Sebastopol, CA : O’Reilly & Associates. 1993.
- [3] Freed N., Borenstein N. “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies.” RFC 2045. Innosoft, First Virtual. 1996.
- [4] Douglas E. Comer **Internetworking With TCP/IP Vol I: Principles, Protocols, and Architecture**. 2<sup>nd</sup> Ed. New Jersey : Prentice-Hall. 1991.
- [5] สรวิศ นุญมี. “โครงสร้างระบบอิเล็กทรอนิกส์ของอินเทอร์เน็ต.” บทความเสนอในการสัมมนาวิชาสัมมนา 1 ระดับปริญญาโท สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2536.
- [6] Postel J. “Simple Mail Transfer Protocol.” STD 10. RFC 821. USC/Information Sciences Institute. 1982.
- [7] Crocker D. “Standard for the Format of ARPA Internet Text Messages.” STD 11. RFC 822. UDEL. 1982.
- [8] Aleks N. “Mailing List Management Software FAQ, v. 1.1.” [Online]. Available : <ftp://ftp.uu.net/usenet/news.answers/mail/list-admin/software-faq>. 1995.
- [9] Thomas E. “Listserv Distribute Protocol.” RFC 1429. Swedish University Network. 1993.
- [10] Sirbu M. “A CONTENT-TYPE HEADER FIELD FOR INTERNET MESSAGES.” RFC 1049. CMU. 1988.
- [11] Freed N., Borenstein N. “Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types.” RFC 2046. Innosoft, First Virtual. 1996.
- [12] Freed N., Moore K. “Definition of the URL MIME External-Body Access-Type.” RFC 2017. Innosoft, University of Tennessee. 1996.
- [13] Berners-Lee T., Masinter L., McCahill M., Editors. “Uniform Resource Locators (URL).” RFC 1738. CERN, Xerox Corporation, University of Minnesota. 1994.
- [14] Alan L. Eliason. **Systems Development: Analysis, Design, and Implementation**. 2<sup>nd</sup> Ed. Illinois : Scott, Foresman and Company. 1989.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [15]ราชบัณฑิตยสถาน. ศัพท์คอมพิวเตอร์. พิมพ์ครั้งที่ 4 แก้ไขเพิ่มเติม. กรุงเทพฯ : ราชบัณฑิตยสถาน. 2540.
- [16]สรวิศ บุญมี. “การแก้ปัญหาของเมลลิงลิสต์โดยใช้ระบบจัดการเมลลิงลิสต์ย่อย.” สารสนเทศลาดกระบัง, ปีที่ 5, ฉบับที่ 1, 2543.
- [17]Borenstein N., Freed N. “MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies.” RFC 1521. Bellcore, Innosoft. 1993.
- [18]Westine A., Postel J. “Problems with the Maintenance of Large Mailing Lists.” RFC 1211. ISI. 1991.
- [19]Gaffin A. “EFF’s Guide to the Internet, v. 3.21.” [Online]. Available : [ftp://ftp.eff.org/pub/Net\\_info/Guidebooks/EFF\\_Net\\_Guide/netguide.eff](ftp://ftp.eff.org/pub/Net_info/Guidebooks/EFF_Net_Guide/netguide.eff). 1999.
- [20]Craig Hunt. **TCP/IP Network Administration**. Sebastopol, CA : O’Reilly & Associates. 1993.
- [21]Larry Wall, Tom Christiansen and Randal L. Schwartz with Stephen Potter. **Programming Perl**. 2<sup>nd</sup> Ed. Sebastopol, CA : O’Reilly & Associates. 1996.
- [22]อัครเสน สมุทรพ่อง, ชงยุทธ จิวุฒิพงศ์ และ วิโรจน์ เด่นวานิช. **คอมพิวเตอร์เบื้องต้น การเขียนโปรแกรมภาษาBASIC**. กรุงเทพฯ : ดาต้าแมท. ม.ป.ป.

## ภาคผนวก ก.

# วิธีใช้ระบบจัดการเมลถึงลิสต์ย่อย

### ความต้องการของระบบ (System Requirement)

- ระบบที่ดำเนินงานโดยระบบปฏิบัติการยูนิกซ์ หรือระบบปฏิบัติการที่เข้ากันได้ เช่น ลินุกซ์
- Mail Transport Agent อะไรก็ได้ ที่สามารถติดต่อได้ด้วยโปรโตคอล SMTP
- เซิร์ฟเวอร์สำหรับบริการข้อมูล เช่น HTTP หรือ FTP เซิร์ฟเวอร์
- Perl รุ่น 5.004 ขึ้นไป
- MailTools 1.11
- IniConf 0.92
- Sprite 3.21
- ซอฟต์แวร์จัดการเมลถึงลิสต์ที่ไม่แก้ไข MIME Header ของจดหมาย เช่น Majordomo หากต้องการความสะดวกในการจัดการเมลถึงลิสต์ย่อย

### การติดตั้ง

1. ทำการสร้างบัญชีผู้ใช้ใหม่ขึ้นมาสำหรับ Sublist Gateway
2. สำเนาเพิ่ม expire.pl gateway.ini gateway.pl list.db และ mid.db จากต้นฉบับไว้ในไดเรกทอรีเดียวกันของระบบ อาจเป็นไดเรกทอรีบ้านของบัญชีผู้ใช้ในข้อ 1 ก็ได้
3. กำหนด Mail Alias ของบัญชีผู้ใช้ในข้อ 1 ให้ส่งจดหมายให้ gateway.pl ทำงานแทน โดยเพิ่มเข้าไปในแฟ้ม aliases หรือ .forward สำหรับ sendmail ดังตัวอย่างต่อไปนี้

```
"|/home/sublist/gateway.pl #sublist"
```

เนื่องจาก sendmail รุ่นก่อน V8 จะเรียงลำดับแอดเดรสทั้งหมด และตัดแอดเดรสที่ซ้ำกันออกก่อนจะส่งจดหมายให้แอดเดรสเหล่านั้น ดังนั้นหากมี Alias ใดใช้โปรแกรม gateway.pl เหมือนกัน โปรแกรมอาจจะไม่ทำงานในทุก Alias ดังนั้นเพื่อหลีกเลี่ยงปัญหาดังกล่าว จึงควรวใส่ Shell Comment เป็นอาร์กิวเมนต์หลอกที่แตกต่างกันในแต่ละ Alias เช่น "#sublist" ดังตัวอย่าง

4. แก้ไขพารามิเตอร์ต่างๆ ในแฟ้ม gateway.ini ตามตารางต่อไปนี้

พารามิเตอร์	ความหมาย	รูปแบบ
MyAddress	อีเมลแอดเดรสของ Sublist Gateway	user@subdomain.subdomain
MailBox	ชื่อแฟ้มของ mailbox ที่ใช้เก็บจดหมายที่ไม่สามารถประมวลผลได้ ถ้าไม่ใส่ path เดิมจะเก็บไว้ในไดเรกทอรีเดียวกับ Sublist Gateway	[/path/path/]mailbox
ArchiveDir	ชื่อไดเรกทอรีที่ใช้เก็บจดหมายที่ส่วนกลางสำหรับสมาชิกของเมลลิงลิสต์ย่อย ถ้าไม่ใส่ path เดิมจะเก็บไว้ในไดเรกทอรีเดียวกับ Sublist Gateway	[/path/path/]archivedir
URLPrefix	URL ส่วนเดิมนำ เพื่อเข้าถึงไดเรกทอรีที่ใช้เก็บจดหมายที่ส่วนกลาง	Scheme://domain/path
Expire	จำนวนวัน ที่จะเก็บจดหมายที่ส่วนกลาง	N (ตัวเลขแสดงจำนวน)

## การใช้งาน

เมื่อต้องการสร้างเมลลิงลิสต์ย่อยให้ทำตามขั้นตอนต่อไปนี้

1. สร้างเมลลิงลิสต์ย่อยตามคำแนะนำของซอฟต์แวร์จัดการเมลลิงลิสต์ที่ใช้
2. เพิ่มอีเมลแอดเดรสของเมลลิงลิสต์หลัก และ เมลลิงลิสต์ย่อยลงในแฟ้ม list.db
3. ส่งจดหมายจากบัญชีผู้ใช้ของ Sublist Gateway ไปสมัครเป็นสมาชิกทั้งเมลลิงลิสต์หลัก และเมลลิงลิสต์ย่อย
4. ตรวจสอบ Mailbox ที่เก็บจดหมายที่ไม่สามารถประมวลผลได้ (ระบุในแฟ้ม gateway.ini) เพื่อตอบยืนยันการเป็นสมาชิกถ้ามี
5. เรียกโปรแกรม expire.pl มาทำงานเป็นระยะๆ โดยอาจใช้ cron ช่วย
6. ตรวจสอบ Mailbox ที่เก็บจดหมายที่ไม่สามารถประมวลผลได้เป็นระยะๆ เพื่อลบจดหมายที่ไม่จำเป็นทิ้ง

## ภาคผนวก ข.

## สรุปคำสั่งและคำตอบรับของโปรโตคอล SMTP

## สรุปคำสั่งของโปรโตคอล SMTP

รูปแบบคำสั่ง	คำอธิบาย
HELO <domain>	ใช้แสดงตัวของฝ่ายส่งต่อฝ่ายรับ เพื่อเริ่มต้นทำรายการต่อไป
MAIL FROM:<reverse-path>	เริ่มทำรายการส่งจดหมาย โดยบอกต้นทางหรือผู้ส่งจดหมาย และส่งจดหมายไปเก็บใน Mailbox
RCPT TO:<forward-path>	แสดงตัวผู้รับปลายทาง ถ้ามีผู้รับหลายคนก็สมารถใช้ได้หลายครั้ง
DATA	เริ่มส่งข้อมูลจดหมาย จบโดยบรรทัดที่มีเพียงจุด (Period)
RSET	ยกเลิกรายการส่งจดหมายที่กำลังทำอยู่
SEND FROM:<reverse-path>	เริ่มทำรายการส่งจดหมาย โดยบอกต้นทางหรือผู้ส่งจดหมาย และส่งจดหมายไปแสดงยังเครื่องปลายทาง (Terminal)
SOML FROM:<reverse-path>	เริ่มทำรายการส่งจดหมาย โดยบอกต้นทางหรือผู้ส่งจดหมาย และส่งจดหมายไปแสดงยังเครื่องปลายทาง (Terminal) หรือเก็บใน Mailbox ถ้าไม่สามารถแสดงยังเครื่องปลายทางได้
SAML FROM:<reverse-path>	เริ่มทำรายการส่งจดหมาย โดยบอกต้นทางหรือผู้ส่งจดหมาย และส่งจดหมายไปแสดงยังเครื่องปลายทาง (Terminal) และเก็บใน Mailbox
VERFY <string>	ยืนยันว่ามีผู้รับตามอาร์กิวเมนต์ที่ระบุ
EXPN <string>	ยืนยันว่าอาร์กิวเมนต์ที่ระบุเป็นเมลลิงลิสต์
HELP [<string>]	ส่งข้อความแสดงความช่วยเหลือให้ฝ่ายส่ง
NOOP	บอกให้ฝ่ายรับไม่ต้องทำอะไรนอกจากตอบรับ
QUIT	ปิดการทำรายการส่งจดหมาย และตัดการติดต่อ
TURN	สลับหน้าที่กันระหว่างฝ่ายส่งและฝ่ายรับ

## รหัสและคำตอบรับ

รหัส	คำตอบรับ
211	System status, or system help reply
214	Help message
220	<domain> Service ready
221	<domain> Service closing transmission channel
250	Requested mail action okay, completed
251	User not local; will forward to <forward-path>
354	Start mail input; end with <CRLF>.<CRLF>
421	<domain> Service not available, closing transmission channel
450	Requested mail action not taken: mailbox unavailable
451	Requested action aborted: local error in processing
452	Requested action not taken: insufficient system storage
500	Syntax error, command unrecognized
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command parameter not implemented
550	Requested action not taken: mailbox unavailable
551	User not local; please try <forward-path>
552	Requested mail action aborted: exceeded storage allocation
553	Requested action not taken: mailbox name not allowed
554	Transaction failed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.  
บทความที่ได้รับการตีพิมพ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การแก้ปัญหาของเมลลิงลิสต์โดยใช้ระบบจัดการเมลลิงลิสต์ย่อย

### Solving Problems of Mailing Lists using Submailing List Management System

สุรสิทธิ์ วรรณไกรโรจน์<sup>1</sup> จันทร์บุรณม์ สถิตวิริยวงศ์<sup>2</sup> และ สรวิศ บุญมี<sup>3</sup>

#### บทคัดย่อ

เมลลิงลิสต์เป็นการประยุกต์ใช้ไปรษณีย์อิเล็กทรอนิกส์โดยใช้ประโยชน์จากความสามารถในการส่งอีเมลเพียงฉบับเดียวถึงผู้รับหลายๆ คน แต่ปัญหาของเมลลิงลิสต์ขนาดใหญ่คือลิสต์ย่อย ซึ่งทำให้เกิดความสับสนในการส่งคำร้องขอ การส่งข้อความแสดงความคิดเห็น และไม่สามารถโต้ตอบกับลิสต์หลักได้ บทความนี้จึงเสนอวิธีแก้ปัญหาเหล่านี้โดยใช้ระบบจัดการเมลลิงลิสต์ย่อยซึ่งทำหน้าที่เสมือนเป็นเกตเวย์สำหรับแลกเปลี่ยนเมลระหว่างลิสต์หลักและลิสต์ย่อย

#### Abstract

Mailing lists represent another way to apply electronic mail by using ability to send an only one message to a list of many recipients. But as lists are getting larger, sublists may cause problems such as misdirected requests, misdirected error reports and can not response with main lists. This paper presents a solution for these problems using Mailing List Management System as a gateway built for message exchange between main lists and sublists.

#### 1. บทนำ

ในบรรดาการประยุกต์ใช้เครือข่ายอินเทอร์เน็ตทั้งหมด ไปรษณีย์อิเล็กทรอนิกส์ หรือเรียกย่อๆ ว่า อีเมล (Electronic mail; e-mail) เป็นการประยุกต์ที่เก่าแก่ดั้งเดิมมาตั้งแต่เริ่มก่อตั้งอินเทอร์เน็ตที่ยังคงมีความสำคัญเป็นที่รู้จัก และมีผู้ใช้งานที่สุดอย่างหนึ่งมาจน

ถึงทุกวันนี้ ผู้คนนับล้านๆ คนทั่วโลกที่ใช้อินเทอร์เน็ตจะต้องมีอีเมลแอดเดรสเป็นของตัวเอง [1]

เมลลิงลิสต์ (Mailing list) หรือ บัญชีจำหน่าย เป็นการประยุกต์ใช้ไปรษณีย์อิเล็กทรอนิกส์รูปแบบหนึ่ง โดยใช้ประโยชน์จากความสามารถในการส่งอีเมลเพียงฉบับเดียวถึงผู้รับหลายๆ คนได้โดยง่ายเพียงจำหน่ายถึงผู้รับเหล่านั้น เมลเซิร์ฟเวอร์ (Mail server) ก็

<sup>1</sup> อาจารย์พิเศษคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

<sup>2</sup> อาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

<sup>3</sup> นักศึกษาปริญญาโท สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

จะทำสำเนาอีเมลฉบับนั้นส่งให้ผู้รับทุกคนโดยอัตโนมัติ ด้วยเหตุนี้จึงมีการนำเมลลิงลิสต์มาประยุกต์ใช้ในการประชุม ปรีกษาหารือ แลกเปลี่ยนความคิดเห็น หรือเพื่อแจ้งข่าวสารถึงกลุ่มสมาชิก

## 2. การจัดการเมลลิงลิสต์

การจัดการเมลลิงลิสต์แบ่งได้เป็นสองแนวทางคือ [2]

1. การจัดการด้วยตัวเองโดยไม่ใช้ซอฟต์แวร์จัดการเมลลิงลิสต์ช่วย
2. การจัดการโดยใช้ซอฟต์แวร์จัดการเมลลิงลิสต์ (Mailing list manager)

การจัดการเมลลิงลิสต์นั้นถ้าสมาชิกที่ใช้ก่อนข้างคงที่ไม่มีการเปลี่ยนแปลงบ่อยนัก ก็อาจไม่จำเป็นต้องใช้ซอฟต์แวร์ช่วยในการจัดการ ถ้าลิสต์นั้นต้องถูกกั้นกรองก่อน (Moderated หรือตรวจสอบความเหมาะสมของเนื้อหาในแต่ละฉบับก่อนที่จะกระจายให้แก่สมาชิก) ก็อาจใช้เพียงแค่อุปกรณ์จัดการอีเมล (Mail User Agent) ทั่วๆ ไปเช่น Pine mh หรือ Eudora ก็เพียงพอต่อการดูแลเมลลิงลิสต์แล้ว เมื่อมีจดหมายมาถึง ก็ใช้โปรแกรมนั้นทำการส่งต่อไปให้สมาชิกในลิสต์ สิ่งที่ควรทำสำหรับแนวทางนี้คือ ถ้าโปรแกรมที่ใช้มีฟังก์ชัน "bounce" ก็ควรจะใช้ฟังก์ชันนี้แทน "forward" เพราะมันจะรักษา Header "From:" ซึ่งแสดงว่าใครเป็นคนส่งต้นฉบับของจดหมาย และแสดงอีเมลแอดเดรสของเจ้าของลิสต์ที่ "Resent-From:" ซึ่งแบบนี้จะเหมาะสมกว่า นอกจากนี้ยังควรใส่รายชื่อของอีเมลแอดเดรสที่ "Bcc:" แทนที่จะเป็น "To:" เพื่อว่าถ้าสมาชิกตอบจดหมายฉบับนั้นมันจะได้ส่งกลับมาที่เจ้าของลิสต์ไม่ใช่ส่งให้ลิสต์ (โดยไม่ผ่านการกั้นกรองจากเจ้าของลิสต์)

นอกจากนี้ยังมีวิธีที่สามารถติดตั้งระบบอัตโนมัติเพื่อการส่งอีเมลกลับให้สมาชิกได้ ถ้าใช้ระบบยูนิกซ์ที่รันโปรแกรม sendmail โดยใช้ฟังก์ชัน "alias" ของ sendmail ซึ่งผู้ดูแลระบบจะต้องติดตั้ง alias พิเศษชนิดหนึ่งให้กับลิสต์ ก็จะสามารถดูแลลิสต์ได้โดยง่ายเพียงแค่ทำการแก้ไขไฟล์ซึ่ง alias นั้นชื่ออยู่เมื่อมีเมลส่งมาที่ alias นี้ เมลฉบับนั้นก็กระจายถึงทั้งลิสต์ อย่างไรก็ตามแนวทางนี้อาจเกิดปัญหาได้มาก เช่น mailing loops และยังไม่สามารถควบคุมว่าใครบ้างที่สามารถส่งเมลให้ลิสต์ ดังนั้นถ้าต้องกระจายอีเมลแบบอัตโนมัติก็ควรเลือกใช้ซอฟต์แวร์จัดการเมลลิงลิสต์

## 3. ซอฟต์แวร์จัดการเมลลิงลิสต์ (Mailing List Manager)

ซอฟต์แวร์จัดการเมลลิงลิสต์คือซอฟต์แวร์ที่ควบคุมลิสต์ของสมาชิก และโดยปกติมักจะควบคุมการกระจายอีเมลให้แก่สมาชิกด้วย ซอฟต์แวร์จัดการเมลลิงลิสต์นั้นมีผู้พัฒนาออกมาให้เลือกใช้ได้หลายตัว อาทิเช่น ListProc LISTSERV Majordomo และ SmartList

### 3.1 ลักษณะสำคัญบางประการของซอฟต์แวร์จัดการเมลลิงลิสต์

หน้าที่พื้นฐานที่สำคัญอย่างหนึ่งของซอฟต์แวร์จัดการเมลลิงลิสต์คือ เตรียมเมลเพื่อให้อ่านสำหรับการแจกจ่ายในวงกว้าง และป้องกันสมาชิกจากบุคคลอื่นที่พยายามจะปลอมตัวเป็นสมาชิก

เมื่อเมลถูกส่งต่อให้ลิสต์ header บางส่วนควรจะถูกลบออก ตัวอย่างเช่น header "Return-receipt-to:" ซึ่งถ้ากระจายผู้ลิสต์อาจสามารถส่งผลให้เกิดเมลตอบรับอัตโนมัติกลับสู่ระบบของผู้ส่งอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ท่วมท้น แนวทางในการกำจัด header เหล่านี้ 2 แนวทางคือ กำจัด header ที่รู้ว่าจะเป็นอันตราย และอนุญาตเฉพาะ header ที่รู้ว่าจะปลอดภัย Majordomo ใช้แนวทางแรกคือตัด header ต้องห้ามในรูทีน “resend” ของมัน ในขณะที่ ListProc ใช้แนวทางตรงกันข้ามคืออนุญาตเฉพาะ header ที่มีรู้จักและตัดที่เหลือทิ้งเกณฑ์ในการพิจารณา header ของซอฟต์แวร์ทั้งสองสามารถแก้ไขได้โดยผู้ดูแลระบบ

บางครั้งซอฟต์แวร์จัดการเมลลิ่งลิสต์ก็ต้องการความแน่ใจว่าบุคคลที่ส่งเมลมาเป็นคนคนนั้นจริงๆ เนื่องจากการปลอมเมลเป็นเรื่องง่าย ดังนั้นบรรทัด “From:” ก็ไม่สามารถเชื่อได้แม้มันจะเป็นหลักฐานเพียงอย่างเดียวที่ซอฟต์แวร์จัดการเมลลิ่งลิสต์ส่วนใหญ่ต้องการยกเว้น

1. ListProc 6.0c ใช้บรรทัด SMTP “From ” แต่สามารถแก้ไขให้ใช้ “From:” เหมือนตัวอื่นได้
2. Majordomo ขอมให้ผู้ส่งเปลี่ยนหลักฐานที่ใช้แสดงตัวได้โดยใช้ header “Reply-To:” เพราะถือว่าการปลอมเมลเป็นเรื่องง่าย ทำไม่ไม่ทำให้การแสดงตัวอย่างถูกต้องเป็นเรื่องที่ง่ายกว่า ซึ่งมีข้อดีคือช่วยทำให้สมาชิกที่ต้องการย้ายอีเมลแอ็ดเดรสสามารถจัดการได้ด้วยตัวเองโดยไม่ต้องขอความช่วยเหลือจากเจ้าของลิสต์
3. SmartList จะอ่านฟิลด์ที่เกี่ยวกับผู้ส่งทั้งหมดรวมทั้ง “From ” “From:” “Sender:” “Reply-To:” และ signature text ถ้ามี แล้วจึงเลือกใช้แล้วแต่สถานการณ์

นอกจากนี้เพื่อป้องกันไม่ให้เกิดการวนซ้ำ (Mailing loop) ซึ่งจะเกิดเมื่อเซิร์ฟเวอร์สองเครื่องต่างถูกกำหนดให้ forward เมลให้แก่กัน โดยไม่รู้ว่ามีอีกฝั่งก็จะ forward เมลนั้นกลับมาให้ การป้องกันทำได้หลายวิธีอาทิเช่น ซอฟต์แวร์จัดการเมลลิ่งลิสต์ส่วนใหญ่จะ

ปฏิเสธเมลที่มี header precedence เป็น “bulk” และจะเติม precedence “bulk” นี้ให้กับเมลที่ส่งไปยังลิสต์เพื่อป้องกันไม่ให้โปรแกรมจำพวก vacation (โปรแกรมอัตโนมัติที่ตอบรับเมลแทน ใช้เมื่อผู้รับไม่สามารถอยู่ตอบเมลได้ทันที) ตอบรับ

#### 4. ปัญหาของเมลลิ่งลิสต์ขนาดใหญ่

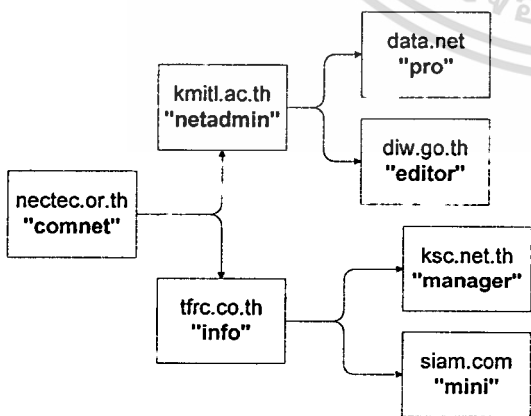
เมลลิ่งลิสต์ขนาดใหญ่อาจได้รับอีเมลถึงประมาณ 400 ฉบับต่อเดือนแสดงความต้องการสมัครเป็นสมาชิกหรือเลิกเป็นสมาชิกของลิสต์เหล่านั้น และมีประมาณ 20 ฉบับต่อวันที่ขอให้แก้ไขลิสต์ [3] นอกจากนี้ยังมีอีเมลที่แสดงถึงความผิดพลาดในการส่งเมลถึงประมาณ 300 ฉบับต่อเดือน เมลพวกนี้ส่วนมากจะซ้ำๆ กัน แต่จะมีสุทธิตั้งแต่วันละประมาณ 10 ฉบับที่ต้องได้รับการตรวจสอบและแก้ไข โดยปัญหาที่พบเป็นประจำมีดังนี้

1. รายงานความผิดพลาดผิดคน (Misdirected Error Reports) ปัญหาที่พบประจำคือเมลรายงานความผิดพลาดจะถูกส่งให้ผู้เขียนเมลแทนที่จะส่งให้กับเจ้าของลิสต์ (List owner) แม้ว่าจะใส่ SMTP “From ” ลงในเมลเมื่อจะกระจายให้แก่ลิสต์แต่เครื่องแม่ข่ายบางเครื่องก็ไม่ได้ยึดถือตามโปรโตคอล SMTP ในการรายงานความผิดพลาด
2. ส่งคำขอร้องผิดที่ (Misdirected Requests) ผู้ใช้บางคนไม่รู้ว่าเมลลิ่งลิสต์มักจะมี “request” mailbox ดังนั้นจึงมักจะหลงส่งคำขอร้องให้กับลิสต์
3. ส่งข้อความผิดที่ (Misdirected Messages) คือเมลที่ส่งมาที่ request mailbox แต่ที่จริงตั้งใจจะส่งให้ลิสต์
4. ลิสต์ย่อย (Sublists) คือ mailbox ที่เป็น alias ซึ่งขยายเป็นเมลลิ่งลิสต์หรือกลุ่มของผู้รับย่อยใน

ลิสต์หลัก บางทีก็เรียกว่า Exploder เพราะมันทำให้ขนาดของลิสต์ (จำนวนผู้รับ) แฉกกระจายออก [4] ดังแสดงในรูปที่ 1 เมื่อมีเมลส่งให้ comnet@nectec.or.th นอกจากจะถูกส่งต่อไปกับสมาชิกตามรายชื่อแล้ว ยังถูกส่งต่อไปให้ netadmin@kmitl.ac.th และ info@tfrc.co.th แต่ผู้รับเหล่านี้ต่างก็เป็นเมลลิ่งลิสต์ และเหมือนลิสต์ comnet ลิสต์เหล่านี้ก็จะส่งให้ผู้ใช้และเมลลิ่งลิสต์อื่น ซึ่งเมลลิ่งลิสต์ขนาดใหญ่ทั้งหลายอาจจะมีหลายลิสต์ย่อย สิ่งที่ผู้ดูแลลิสต์ย่อยเหล่านั้นควรทำคือ กำหนดความเป็นเจ้าของ (List ownership) สำหรับลิสต์ย่อยที่แต่ละ site ของลิสต์ย่อยนั้น เพื่อว่าเวลาเกิดความผิดพลาดอันเกิดจากลิสต์ย่อยนั้น ข้อความแสดงความคิดเห็นจะได้ส่งถึงเจ้าของลิสต์ย่อย ดังตัวอย่างต่อไปนี้แสดงรายการหนึ่งในไฟล์ aliases เพื่อกำหนดเจ้าของของลิสต์ ietf-local

owner-ietf-local: stev@vax.ftp.com

แต่ก็อาจเกิดความไม่เข้าใจหรือกำหนดไม่ถูกต้องทำให้เกิดปัญหาต่างๆ เช่น "User unknown" ที่ไม่ได้เกิดจากลิสต์หลัก หรือผู้ใช้ส่งคำร้องขอเปลี่ยน



รูปที่ 1 Sublist หรือ Exploder ของ Mailing list

แปลงต่างๆ มาที่ Request mailbox ของลิสต์หลัก แต่ที่จริงผู้ใช้นั้นอยู่ในลิสต์ย่อยอันใดอันหนึ่ง ซึ่งบางทีก็ยากที่จะสืบทราบ

นอกจากนี้ตามปกติแล้วลิสต์ย่อยจะถูกสร้างขึ้นจาก alias ซึ่งในกรณีที่ผู้รับข่าวสารเพียงอย่างเดียวก็จะมีปัญหาอะไร แต่ถ้าสมาชิกในลิสต์ย่อยต้องการโต้ตอบหรือเป็นฝ่ายส่งข่าวสารต่างๆ ให้ลิสต์หลักแล้วอาจจะเกิดข้อจำกัดขึ้นได้เนื่องจากสมาชิกในลิสต์ย่อยเหล่านั้นไม่ได้มีรายชื่ออยู่ในลิสต์หลัก ดังนั้นจึงอาจถูกปฏิเสธซึ่งก็แล้วแต่นโยบายของลิสต์นั้นๆ คงอาจทำได้เพียงส่งข้อมูลโต้ตอบกันได้ภายในลิสต์ย่อยทำให้ทิศทางในการสื่อสารถูกจำกัดลง

5. ความซ้ำซ้อนในการส่งและจัดเก็บ เนื่องจากระบบเมลลิ่งลิสต์จะส่งเมลถึงผู้ใช้ทุกคน แม้ว่าผู้ใช้เหล่านั้นจะอยู่บนเครือข่ายย่อย โดเมน หรือแม่ข่ายเดียวหรือใกล้เคียงกัน ทำให้เกิดความซ้ำซ้อนของข้อมูล โดยเฉพาะอย่างยิ่งถ้ามีสมาชิกของลิสต์เดียวกันอยู่ภายในเครือข่ายใกล้เคียงกันเป็นจำนวนมาก ทำให้การจราจรระหว่างเครือข่ายสูงขึ้นโดยไม่จำเป็น นอกจากนี้ยังต้องสิ้นเปลืองเนื้อที่ในการเก็บเมลเหล่านี้ในสมาชิกแต่ละคน ซึ่งส่วนใหญ่จะไม่สามารถอ่านเมลจากเมลลิ่งลิสต์ที่มีสมาชิกจำนวนมากหมดได้ในวันเดียว ทำให้เกิดการสะสมของเมลที่ซ้ำซ้อนกันเป็นจำนวนมากใน mailbox ของแต่ละคน [5]

5. แนวทางการแก้ปัญหา

จากปัญหาดังกล่าว ในข้อ 1 ถึง 3 สามารถแก้ไขได้โดยใช้ซอฟต์แวร์จัดการเมลลิ่งลิสต์ ส่วนปัญหาข้อ 5 แนวทางการแก้ปัญหาคือใช้เนื้อที่จัดเก็บซ้ำซ้อนกันนั้น สามารถแก้ไขได้โดยใช้ลักษณะการจัดเก็บเช่น

เดียวกับ Usenet กล่าวคือ เก็บเมลแต่ละก๊อปปี้ไว้เพียง  
ที่เดียวสำหรับเครือข่ายที่ใกล้เคียงกัน เมื่อผู้ใช้ต้องการ  
อ่านเมลก็ค่อยจัดการดึงเมลที่เก็บไว้มาแสดง จะเห็น  
ได้ว่าการจะทำเช่นนี้ได้ ตัว Mail user agent หรือ  
โปรแกรมที่แสดงเมลแก่ผู้ใช้จะต้องรู้จักการจัดเก็บ  
แบบนี้ และต้องสามารถดึงมาใช้ได้ ซึ่งคงไม่เหมาะสม  
ถ้าต้องตามแก้ไข User agent ทุกตัวให้ทำเช่นนี้ได้ ทาง  
ออกคือใช้โพรโทคอล MIME (Multipurpose Internet  
Mail Extensions) ซึ่งขณะที่เขียนบทความนี้อยู่โพรโท  
คอลนี้มีสถานะเป็น Standards Track ซึ่งกำลังจะเป็น  
มาตรฐานแล้ว ถึงแม้ว่าโพรโทคอลนี้จะยังไม่เป็น  
มาตรฐานเต็มตัวแต่ก็มีความสมบูรณ์ค่อนข้างสูง ทำ  
ให้ Mail user agent ส่วนใหญ่ในขณะนี้สนับสนุนโพร  
โทคอลนี้ และเชื่อแน่ว่าเมื่อโพรโทคอลนี้ได้รับการ  
ยอมรับเป็นมาตรฐานสำหรับอินเทอร์เน็ตแล้ว Mail  
user agent ทุกตัวบนอินเทอร์เน็ตก็จะสนับสนุนโพร  
โทคอลนี้ในที่สุด

โดยใช้ Media type แบบ Message Content-  
Type คือเนื้อความแบบที่บรรจุเมลไว้ภายในอีกที  
(encapsulate เมลอื่นไว้ภายใน) และใช้ subtype แบบ  
external-body ซึ่งแสดงว่าข้อมูลที่เป็นส่วน body ของ  
เมลที่ถูก encapsulate ไม่ได้ถูกรวมไว้ คงมีเพียงแต่  
ส่วนที่ใช้อ้างอิงถึงเท่านั้น โดยจะมีพารามิเตอร์ที่  
อธิบายกลไกในการเข้าถึงข้อมูลที่อยู่ภายนอกอีกที [6]

พารามิเตอร์ที่บังคับเพียงหนึ่งเดียวของ  
message/external-body คือ "access-type" ซึ่งแสดงกล  
ไกในการเข้าถึง body จริงๆ ของ encapsulated mail  
โดย access-type ที่ใช้คือแบบ "URL" ซึ่งแสดงว่า  
body จริงๆ จะถูกเข้าถึงได้จาก URL ที่กำหนดใน  
พารามิเตอร์ URL [7] ตัวอย่างเช่น

```
URL="http://www.foo.com/file"
```

```
Content-Type: message/rfc822  
Content-Id: <id@me.com>
```

```
THIS IS NOT REALLY THE BODY!
```

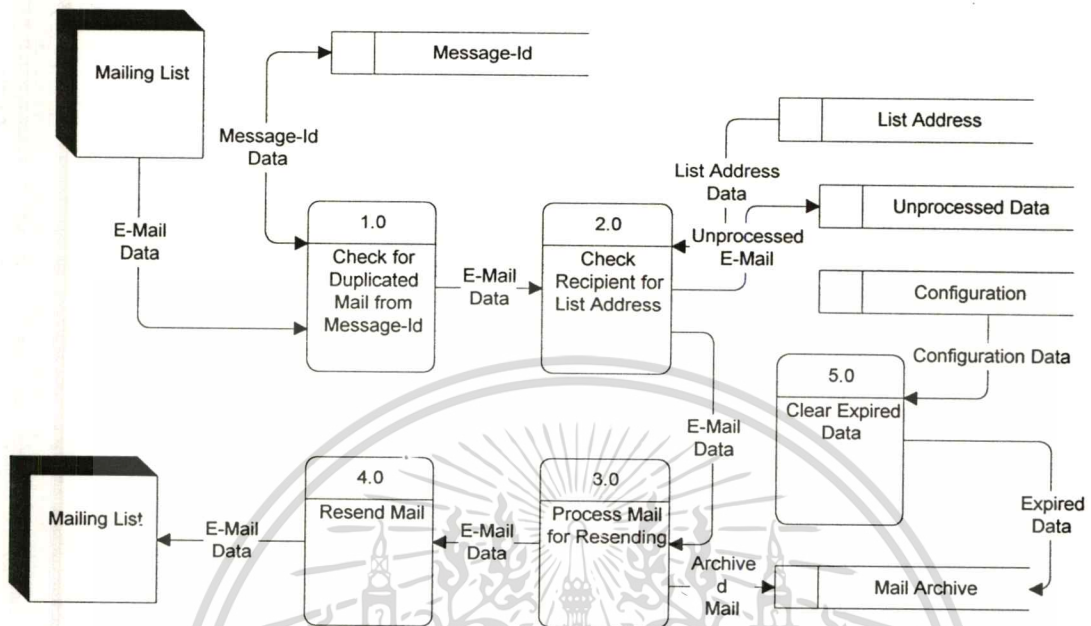
จะเห็นได้ว่าการจัดเก็บและส่งเมลตามโพร  
โทคอล MIME นี้จะช่วยลดเนื้อที่ที่ใช้เก็บเมลที่ซ้ำ  
ซ้อนกันลงไปได้ เพราะ body ของเมลจริงๆ จะเก็บอยู่  
ที่แม่ข่ายส่วนกลางเพียงสำเนาเดียว สำหรับเมลที่ส่ง  
ให้ผู้รับแต่ละคนจะเป็นเพียงส่วนที่อ้างอิงกลับไปหา  
body จริงๆ ที่ส่วนกลางนี้เท่านั้น ซึ่งจะมีเพียง header  
ขนาดเพียงไม่กี่ไบต์ซึ่งจะเป็นการประหยัดเนื้อที่ใน  
การเก็บเมลลงไปได้มากเมื่อเทียบกับการที่ต้องส่ง  
body จริงๆ ที่อาจมีขนาดหลายพันไบต์ไปให้ผู้รับทุก  
คน

สำหรับปัญหาความซ้ำซ้อนในการส่งนั้น  
สามารถแก้ได้โดยใช้ลิสต์ย่อยสำหรับเครือข่ายที่ใกล้  
เคียงกัน เมื่อมีเมลมาลิสต์หลักก็จะส่งมาให้ลิสต์ย่อย  
เพียงก๊อปปี้เดียวจากนั้นลิสต์ย่อยจึงมากระจายต่อให้  
สมาชิกในเครือข่ายใกล้เคียง ส่วนปัญหาต่างๆ ของ  
ลิสต์ย่อย เช่น การทำให้สามารถโต้ตอบกับลิสต์หลัก  
ได้นั้นแก้ไขได้โดยระบบจัดการเมลลิ่งลิสต์ย่อย

## 6. ระบบจัดการเมลลิ่งลิสต์ย่อย

ระบบจัดการเมลลิ่งลิสต์ย่อย (Submailing  
List Management System) เป็นระบบที่จะนำแนวทาง  
ในการแก้ปัญหาดังกล่าวข้างต้นมาปฏิบัติ โดยแสดง  
Dataflow diagram ของระบบได้ดังรูปที่ 2 ซึ่งจาก  
Dataflow diagram สามารถแบ่งการทำงานออกได้เป็น  
สามมอดูล (Module) คือ ลิสต์ย่อย Sublist gateway  
(ประกอบด้วยโพรเซสที่ 1.0 2.0 3.0 และ 4.0) และ  
Maintenance module (โพรเซสที่ 5.0)

```
Content-type: message/external-body;  
access-type=URL;
```



รูปที่ 2 Dataflow Diagram ของระบบจัดการเมลลิ่งลิสต์ย่อย

6.1 ลิสต์ย่อย

ในส่วนของลิสต์ย่อยนั้นสามารถที่จะใช้ซอฟต์แวร์จัดการเมลลิ่งลิสต์ตัวใดมาจัดการก็ได้ ซึ่งจะทำให้ปัญหาในการส่งข้อความแสดงความคิดเห็นและความสับสนจากการส่งข้อความของผู้ใช้หมดไป เนื่องจากซอฟต์แวร์เหล่านี้จะคอยระวังป้องกันให้อยู่แล้ว ส่วนปัญหาในการติดต่อระหว่างลิสต์นั้นตัว Sublist gateway จะแก้ไขได้

6.2 Sublist gateway

จากลักษณะของซอฟต์แวร์จัดการเมลลิ่งลิสต์ที่จะอนุญาตให้เฉพาะสมาชิกเท่านั้นที่จะส่งเมลให้ลิสต์ได้ดังที่กล่าวมาข้างต้น ทำให้โดยปกติแล้วจึงเป็นไปได้ที่จะนำซอฟต์แวร์จัดการเมลลิ่งลิสต์มาจัดการลิสต์ย่อยในลิสต์หลักที่ใช้ซอฟต์แวร์จัดการเมลลิ่งลิสต์จัดการอยู่ เพราะสมาชิกของลิสต์หลักไม่ได้เป็นสมาชิกของลิสต์ย่อยและสมาชิกของลิสต์ย่อยก็ไม่

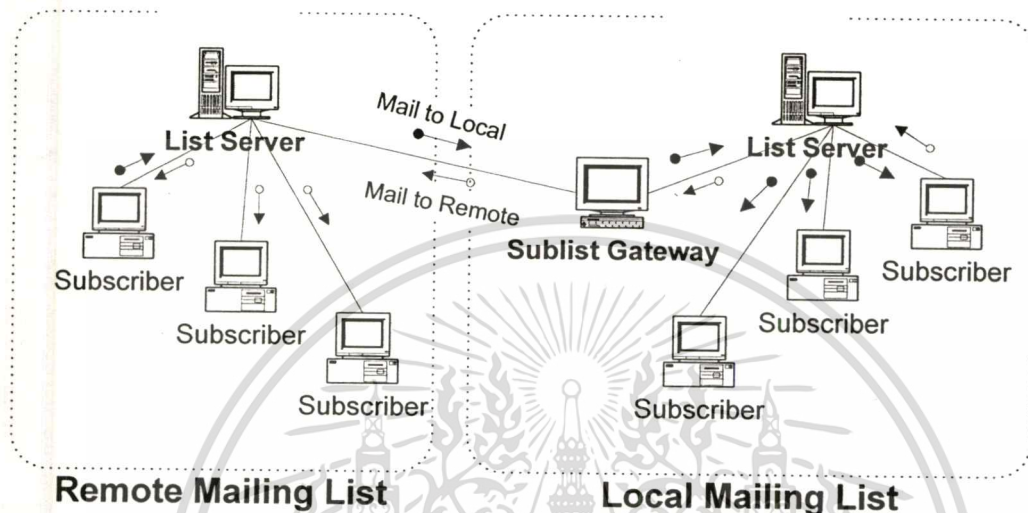
ได้เป็นสมาชิกของลิสต์หลัก ดังนั้นจึงไม่สามารถส่งเมลข้ามระหว่างลิสต์ได้

ดังนั้นวิธีแก้ไขคือสร้างซอฟต์แวร์ที่ทำหน้าที่เป็นเกตเวย์ที่ใช้เชื่อมเมลลิ่งลิสต์หลักและเมลลิ่งลิสต์ย่อยเข้าด้วยกัน โดยเกตเวย์นี้จะจำลองตัวเองเสมือนเป็นสมาชิกธรรมดาคนหนึ่งของทั้งลิสต์หลักและลิสต์ย่อย และจะมีหน้าที่คอยแลกเปลี่ยนเมลระหว่างทั้งสองลิสต์

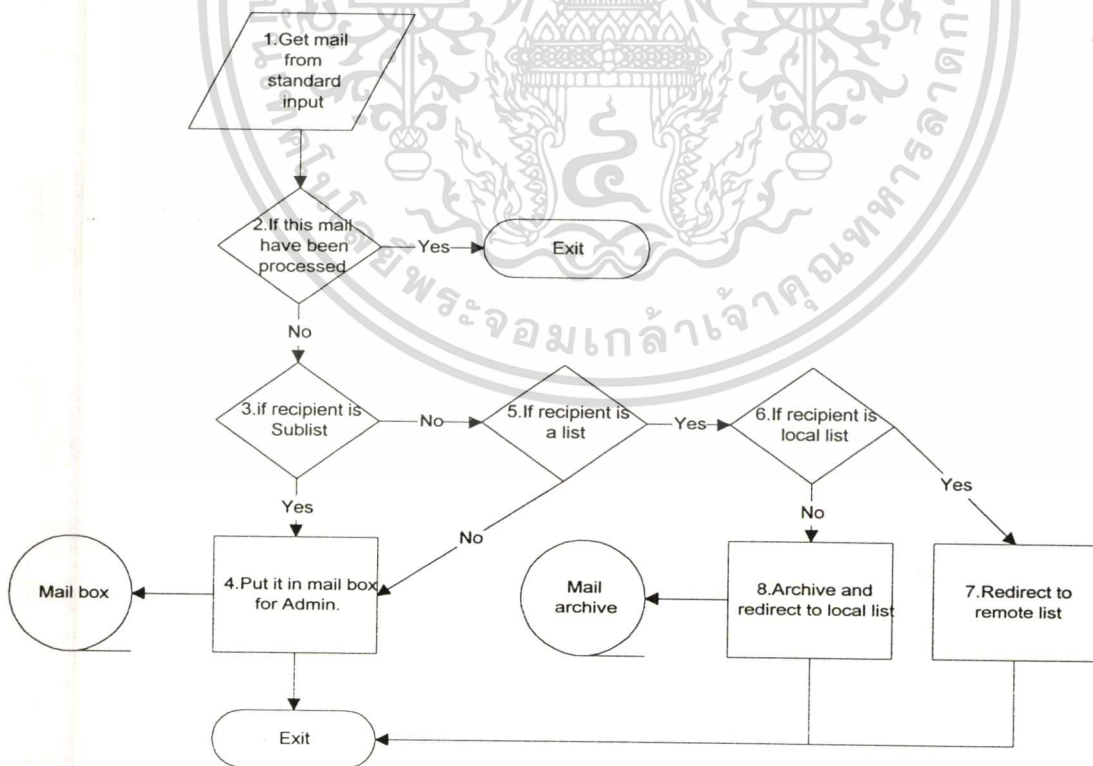
เกตเวย์นี้จะทำงานบนระบบปฏิบัติการใดก็ได้ที่เป็นแบบเดียวกับยูนิกซ์ เช่น ลินุกซ์ โดยใช้ภาษา Perl ในการพัฒนา และติดต่อกับ Mail transport agent ด้วยโปรโตคอล SMTP โดยในขั้นแรก ให้สร้างบัญชีผู้ใช้ให้เกตเวย์ และกำหนด Mail alias ของบัญชีผู้ใช้นี้ให้ชี้ไปที่เกตเวย์เพื่อให้มันจัดการเมลแทน จากนั้นก็ใช้บัญชีผู้ใช้นี้ส่งอีเมลไปสมัครเมลลิ่งลิสต์ทั้งลิสต์หลักและลิสต์ย่อย และบันทึกแอดเดรสของเมลลิ่งลิสต์เหล่านั้นไว้ในฐานข้อมูลของเกตเวย์เพื่อให้มันจัดการต่อไป

รูปที่ 3 แสดงทิศทางในการถ่ายทอจดเมลได้  
ตอบกันระหว่างลิสต์หลักที่อยู่ห่างไกลกับลิสต์ย่อยที่  
อยู่ภายในเครือข่ายใกล้เคียง

เมื่อมีเมลที่ส่งให้ลิสต์เข้ามา เกตเวย์นี้จะ  
ทำงานตามขั้นตอนดังรูปที่ 4 โดยแต่ละบล็อกอธิบาย  
ได้ดังนี้



รูปที่ 3 ทิศทางในการถ่ายทอจดเมล



รูปที่ 4 Flowchart แสดงการทำงานของ Sublist Gateway

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. From lp-l@sor.tfrc.co.th Sun Jun 14 14:58:03 1998  
 2. Return-Path: <lp-l@sor.tfrc.co.th>  
 3. Received: from sor (server@localhost [127.0.0.1])  
 4. by sor.tfrc.co.th (8.8.5/8.8.5) with SMTP id OAA00318  
 5. for <user@sor.tfrc.co.th>; Sun, 14 Jun 1998 14:58:02 +0700  
 6. Date: Sun, 14 Jun 1998 14:58:02 +0700  
 7. Message-Id: <Pine.LNX.3.96.980614145703.279A-100000@sor.tfrc.co.th>  
 8. Errors-To: sor@sor.tfrc.co.th  
 9. Reply-To: lp-l@sor.tfrc.co.th  
 10. Originator: lp-l@sor.tfrc.co.th  
 11. Sender: lp-l@sor.tfrc.co.th  
 12. Precedence: none  
 13. From: sublist@sor.tfrc.co.th  
 14. To: Multiple recipients of list <lp-l@sor.tfrc.co.th>  
 15. Subject: forteenth  
 16. X-Listprocessor-Version: 6.0c -- ListProcessor by Anastasios Kotsikonas  
 17. Mime-Version: 1.0  
 18. Content-Type: message/rfc822  
 19. X-Mailer: Perl5 Mail::Internet v1.30  
 20. >From owner-mj-l@sor.tfrc.co.th Sun Jun 14 14:57:48 1998  
 21. Received: (from majordom@localhost) by sor.tfrc.co.th (8.8.5/8.8.5) id  
 22. OAA00291 for mj-l-list; Sun, 14 Jun 1998 14:57:48 +0700  
 23. X-Authentication-Warning: sor.tfrc.co.th: majordom set sender to  
 24. owner-mj-l@sor.tfrc.co.th using -f  
 25. Received: from localhost (test@localhost) by sor.tfrc.co.th (8.8.5/8.8.5)  
 26. with SMTP id OAA00288 for <mj-l@sor.tfrc.co.th>; Sun, 14 Jun 1998  
 27. 14:57:45 +0700  
 28. Date: Sun, 14 Jun 1998 14:57:45 +0700 (ICT)  
 29. From: For Testing <test@sor.tfrc.co.th>  
 30. To: mj-l@sor.tfrc.co.th  
 31. Subject: forteenth  
 32. Message-Id: <Pine.LNX.3.96.980614145703.279A-100000@sor.tfrc.co.th>  
 33. Mime-Version: 1.0  
 34. Content-Type: TEXT/PLAIN; charset=US-ASCII  
 35. Sender: owner-mj-l@sor.tfrc.co.th  
 36. Precedence: bulk  
 37. This is the actual body of the 14th mail ....

เมลฉบับนี้แสดงการส่งต่อโดยเกิดเว็จาก โดยจะยังรักษาข้อมูลของต้นฉบับไว้เช่น ผู้ส่งที่แท้จริง  
 ลิสต์ย่อยที่ใช้ Majordomo ไปลิสต์หลักที่ใช้ ListProc คือ test@sor.tfrc.co.th (บรรทัดที่ 29) ส่วน body ที่แท้  
 โดยจากบล็อกที่ 7 ของรูปที่ 4 เมลต้นฉบับ (บรรทัดที่ 20 - 37) จะถูกห่อหุ้มตามมาตรฐาน MIME โดยมี ส่วนตัวอย่างต่อไปเป็นเมลที่ส่งต่อจากลิสต์  
 Content-Type แบบ message/rfc822 (บรรทัดที่ 18) หลักมาลิสต์ย่อย

1. From owner-mj-l@sor.tfrc.co.th Tue Feb 29 15:31:43 2000  
 2. Return-Path: <owner-mj-l@sor.tfrc.co.th>  
 3. Received: (from majordom@localhost)  
 4. by sor.tfrc.co.th (8.8.5/8.8.5) id PAA00571  
 5. for mj-l-list; Tue, 29 Feb 2000 15:31:43 +0700  
 6. X-Authentication-Warning: sor.tfrc.co.th: majordom set sender to

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. owner-mj-1@sor.tfrc.co.th using -f  
 8. Received: from sor.tfrc.co.th (sublist@localhost [127.0.0.1])  
 9. by sor.tfrc.co.th (8.8.5/8.8.5) with SMTP id PAA00568  
 10. for <mj-1@sor.tfrc.co.th>; Tue, 29 Feb 2000 15:31:41 +0700  
 11. Date: Tue, 5 May 1998 01:48:06 +0700 (ICT)  
 12. From: sublist@sor.tfrc.co.th  
 13. Reply-To: Sorawit <sor@sor.tfrc.co.th>  
 14. Subject: Hi!  
 15. Message-Id: <Pine.LNX.3.96.980505014710.184D-100000@sor.tfrc.co.th>  
 16. Mime-Version: 1.0  
 17. Content-Type: message/external-body; access-type=URL;  
 18. URL="http://localhost/sublist/lp-1\*40sor.tfrc  
 19. .co.th/Pine.LNX.3.96.980505014710.184D-1 00000\*40sor.tfrc.co.th"  
 20. To: mj-1@sor.tfrc.co.th  
 21. X-Mailer: Perl5 Mail::Internet v1.30  
 22. Sender: owner-mj-1@sor.tfrc.co.th  
 23. Precedence: bulk  
 24. Content-Type: message/rfc822  
 25. Content-Id: <SL.Pine.LNX.3.96.980505014710.184D-100000@sor.tfrc.co.th>

จากการทำงานตามบล็อกที่ 8 ของรูปที่ 4 จะเห็นว่าเมลฉบับนี้ไม่มี body ที่แท้จริง ในบรรทัดที่ 24 เป็นต้นไปนั้นเป็น header ของ encapsulated message ส่วน body ที่แท้จริงจะสามารถเข้าถึงได้จาก URL ที่ระบุในบรรทัดที่ 18

จากการทดลองจัดการลิสต์ย่อยของเมลลิ่งลิสต์ Communication Network (commnet@nectec.or.th) ซึ่งมีขนาดเมลเฉลี่ย 2,948 ไบต์/ฉบับ เมื่อใช้ระบบจัดการเมลลิ่งลิสต์ย่อยเปรียบเทียบกับแบบที่ไม่ใช้ ปริมาณข้อมูลของเมลที่ส่งมาจากเมลลิ่งลิสต์หลักจะลดลง 50% เมื่อมีสมาชิกในลิสต์ย่อย 2 คน และจะลดได้มากขึ้นเมื่อสมาชิกในลิสต์ย่อยมีมากขึ้น นั่นคืออาจลดได้ถึง 80% เมื่อมีสมาชิกในลิสต์ย่อย 5 คน ส่วนการใช้เนื้อที่ของแม่ข่ายในการเก็บเมลจากลิสต์หลักของสมาชิกในลิสต์ย่อยจะลดลง 13% เมื่อมีสมาชิกในลิสต์ย่อย 2 คน และจะลดได้มากขึ้นเมื่อสมาชิกในลิสต์ย่อยมีมากขึ้น นั่นคืออาจลดได้ถึง 43% เมื่อมีสมาชิกในลิสต์ย่อย 5 คน

## 8. สรุป

ระบบจัดการเมลลิ่งลิสต์ย่อยจะช่วยกระจายภาระในการจัดการและดูแลเมลลิ่งลิสต์ขนาดใหญ่ และช่วยลดการจราจรระหว่างเครือข่ายเนื่องจากเสมือนเป็นการจำลองลิสต์หลักมา ซึ่งสามารถให้ผู้ใช้ที่อยู่ในเครือข่ายใกล้เคียงกับลิสต์ย่อยมาสมัครและรับเมลจากลิสต์ย่อยแทนโดยไม่ต้องสนใจลิสต์หลัก ทำให้ลดจำนวนเมลจากลิสต์หลักมาสู่ลิสต์ย่อยและเครือข่ายใกล้เคียงได้ นอกจากนี้ยังช่วยลดปัญหาการจุกเกี่ยวกับเมลที่ซ้ำซ้อนกันของสมาชิกแต่ละคนในลิสต์เดียวกัน และแก้ปัญหาเมลเก่าๆ ที่ตกค้างสะสมอยู่ได้โดยการลบทิ้งเป็นระยะตามอายุของเมลนั้น

เนื่องจากเมลที่ส่งผ่านระบบจัดการเมลลิ่งลิสต์ย่อยต้องอาศัยรูปแบบตามโพรโทคอล MIME ดังนั้นจึงเป็นข้อจำกัดที่อาจทำให้ไม่สามารถใช้งานได้ถูกต้องกับซอฟต์แวร์จัดการเมลลิ่งลิสต์หรือ Mail user agent รุ่นเก่าที่ยังไม่สนับสนุนโพรโทคอลนี้ นอกจากนี้แม้ Mail user agent รุ่นใหม่บางตัวก็ไม่สนับสนุนโพรโทคอล MIME อย่างสมบูรณ์ ดังนั้นข้อเสนอแนะสำหรับการศึกษาคือ ไปได้คือหาวิธีอื่นเพิ่มเติมที่จะช่วย

ให้ Mail user agent พวกนี้ดึงเมลจากส่วนกลางไป  
อ่านได้อย่างอัตโนมัติ

### เอกสารอ้างอิง

- [1] Gaffin A. "EFF's Guide to the Internet, v. 3.21." [Online]. Available : [ftp://ftp.eff.org/pub/Net\\_info/Guidebooks/EFF\\_Net\\_Guide/netguide.eff](ftp://ftp.eff.org/pub/Net_info/Guidebooks/EFF_Net_Guide/netguide.eff). 1999.
- [2] Aleks N. "Mailing List Management Software FAQ, v. 1.1." [Online]. Available : <ftp://ftp.uu.net/usenet/news.answers/mail/list-admin/software-faq>. 1995.
- [3] Westine A., Postel J. "Problems with the Maintenance of Large Mailing Lists." RFC 1211. ISI. 1991.
- [4] Bryan Costales, Eric Allman, Neil Rickert. **sendmail**. Sebastopol, CA : O'Reilly & Associates, Inc. 1993.
- [5] สรวีศ บุญมี. "ระบบจัดการเมลถึงลิสต์ย่อย." บทความเสนอในการสัมมนาวิชาสัมมนา 2 ระดับปริญญาโท สาขาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2537.
- [6] Freed N., Borenstein N. "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types." RFC 2046. Innosoft, First Virtual. 1996.
- [7] Freed N., Moore K. "Definition of the URL MIME External-Body Access-Type." RFC 2017. Innosoft, University of Tennessee. 1996.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

นายสรวิศ บุญมี เกิดเมื่อวันที่ 11 กันยายน 2512 ที่จังหวัดกรุงเทพฯ สำเร็จการศึกษาวិทยาศาสตร์บัณฑิต (คณิตศาสตร์ประยุกต์) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2534

ปี พ.ศ. 2535 เข้าทำงานในบริษัทด้าเมท จำกัด ในตำแหน่งนักเขียนโปรแกรม

ปี พ.ศ. 2535 เข้าทำงานในบริษัทกรุงเทพอินเอกซ์ จำกัด ในเครือเจริญโภคภัณฑ์ ในตำแหน่งนักเขียนโปรแกรม

ปี พ.ศ. 2538 เข้าทำงานในบริษัทศูนย์วิจัยกสิกรไทย จำกัด ในตำแหน่งเจ้าหน้าที่คอมพิวเตอร์จนถึงปัจจุบัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้